



Docket No. 1080.1045CIPD3

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Fumio NAGASHIMA

Application No.: 09/765,380

Group Art Unit: 2122

Filed: January 22, 2001

Examiner: Vo, Ted T.

For: METHOD FOR STORING SOFTWARE OBJECTS FROM EXISTING SOFTWARE
COMPONENTS (AS AMENDED)

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith
a certified copy of the following foreign application:

Japanese Patent Application No(s). 9-171782

Filed: June 27, 1997

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 8 2006

By: Mark J. Henry
Mark J. Henry
Registration No. 36,162

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 1 9 9 7 年 6 月 2 7 日
Date of Application:

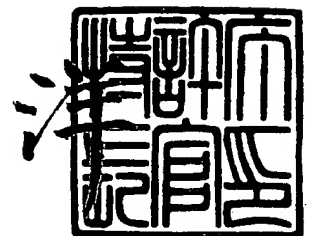
出 願 番 号 平成 9 年特許願第 1 7 1 7 8 2 号
Application Number:
[ST. 10/C]: [J P 1 9 9 7 - 1 7 1 7 8 2]

願 人 富士通株式会社
Applicant(s):

2 0 0 5 年 3 月 7 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



【書類名】 特許願

【整理番号】 9703799

【提出日】 平成 9年 6月27日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/00
G06F 9/06

【発明の名称】 オブジェクト指向プログラミング装置、オブジェクト指向プログラミング支援装置、コンポーネント作成装置、オブジェクト指向プログラム記憶媒体、オブジェクト指向プログラミング用プログラム記憶媒体、コンポーネント記憶媒体、およびオブジェクト間ネットワーク表示方法

【請求項の数】 53

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 永嶋 史朗

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 植木 美和

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 遠藤 香緒里

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 須田 良輔

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 湯本 麻子

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 丸山 次人

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 佐々木 繁

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100094330

【弁理士】

【氏名又は名称】 山田 正紀

【先の出願に基づく優先権主張】

【出願番号】 平成 8 年特許願第 170079 号

【出願日】 平成 8 年 6 月 28 日

【先の出願に基づく優先権主張】

【出願番号】 平成 8 年特許願第 292863 号

【出願日】 平成 8 年 11 月 5 日

【手数料の表示】

【予納台帳番号】 017961

【納付金額】 21,000 円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704376

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 オブジェクト指向プログラミング装置、オブジェクト指向プログラミング支援装置、コンポーネント作成装置、オブジェクト指向プログラム記憶媒体、オブジェクト指向プログラミング用プログラム記憶媒体、コンポーネント記憶媒体、およびオブジェクト間ネットワーク表示方法

【特許請求の範囲】

【請求項1】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、該第1のオブジェクトのメッセージと該第2のオブジェクトのメソッドとを対応づけることにより、該第1のオブジェクトと該第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されてなるデータ要素リストを生成するデータ要素リスト生成手段と、

オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されてなるポインタ要素リストを生成するポインタ要素リスト生成手段と、

データ要素リストを持つ第3のオブジェクトとポインタ要素リストを持つ第4のオブジェクトとの間で、該第4のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、該第3のオブジェクトのデータ要素リストに配列されたポインタを書き込むことにより、該第3のオブジェクトと該第4のオブジェクトとの間でのデータの授受を可能ならしめるデータ結合手段とを備えたことを特徴とするオブジェクト指向プログラミング装置。

【請求項2】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、該メソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタと

を含むメソッド要素の配列であるメソッド要素リストを生成するものであることを特徴とする請求項 1 記載のオブジェクト指向プログラミング装置。

【請求項 3】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第 1 のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第 2 のオブジェクトとの間で、該第 1 のオブジェクトのメッセージと該第 2 のオブジェクトのメソッドとを対応づけることにより、該第 1 のオブジェクトと該第 2 のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、該他オブジェクトのメッセージと自オブジェクトのメソッドとの対応をあらわす入力インストラクションタグテーブルを生成する入力インストラクションタグテーブル生成手段とを備えたことを特徴とするオブジェクト指向プログラミング装置。

【請求項 4】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッド ID と、該メソッド ID により特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

前記入力インストラクションタグテーブル生成手段が、前記入力インストラクションタグテーブルを生成するとともに、該入力インストラクションタグテーブルを、該入力インストラクションタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることを特徴とする請求項 3 記載のオブジェクト指向プログラミング装置。

【請求項 5】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第 1 のオブジェクトと、他オブジェクトで発行された自オブジェ

クト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、該第1のオブジェクトのメッセージと該第2のオブジェクトのメソッドとを対応づけることにより、該第1のオブジェクトと該第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、該他オブジェクトのメソッドと自オブジェクトのメッセージとの対応をあらわす出力インストラクションタグテーブルを生成する出力インストラクションタグテーブル生成手段とを備えたことを特徴とするオブジェクト指向プログラミング装置。

【請求項6】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、該メソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

前記出力インストラクションタグテーブル生成手段が、前記出力インストラクションタグテーブルを生成するとともに、該出力インストラクションタグテーブルを、該出力インストラクションタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることを特徴とする請求項5記載のオブジェクト指向プログラミング装置。

【請求項7】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、該第1のオブジェクトのメッセージと該第2のオブジェクトのメソッドとを対応づけることにより、該第1のオブジェクトと該第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれ

に対応する、該他オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リスト ID と、自オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リスト ID との対応をあらわす入力データタグテーブルを生成する入力データタグテーブル生成手段とを備えたことを特徴とするオブジェクト指向プログラミング装置。

【請求項 8】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッド ID と、該メソッド ID により特定されるメソッドが実行される他のオブジェクトへのポインタを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

前記入力データタグテーブル生成手段が、前記入力データタグテーブルを生成するとともに、該入力データタグテーブルを、該入力データタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることを特徴とする請求項 7 記載のオブジェクト指向プログラミング装置。

【請求項 9】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第 1 のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第 2 のオブジェクトとの間で、該第 1 のオブジェクトのメッセージと該第 2 のオブジェクトのメソッドとを対応づけることにより、該第 1 のオブジェクトと該第 2 のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、該他オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リスト ID と、自オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リスト ID との対応をあらわす出力データタグテーブルを生成する出力データタグテーブル生成手段とを

備えたことを特徴とするオブジェクト指向プログラミング装置。

【請求項 10】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する、他オブジェクトのメソッドを特定するメソッド ID と、該メソッド ID により特定されるメソッドが実行される第 2 のオブジェクトへのポインタを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

前記出力データタグテーブル生成手段が、前記出力データタグテーブルを生成するとともに、該出力データタグテーブルを、該出力データタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることを特徴とする請求項 9 記載のオブジェクト指向プログラミング装置。

【請求項 11】 データと処理とを有する複数のオブジェクトが格納されるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第 1 のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第 2 のオブジェクトとの間で、該第 1 のオブジェクトのメッセージと該第 2 のオブジェクトのメソッドとを対応づけることにより、該第 1 のオブジェクトと該第 2 のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されてなるデータ要素リストを生成するデータ要素リスト生成手段と、

オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されてなるポインタ要素リストを生成するポインタ要素リスト生成手段と、

データ要素リストを持つ第 3 のオブジェクトとポインタ要素リストを持つ第 4 のオブジェクトとの間で、該第 4 のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、該第 3 のオブジェクトのデータ要素リストに配列されたポインタを書き込むことにより、該第 3 のオブジェクトと該第 4 のオブジェクトとの間でのデータの授受を可能ならしめるデータ結合手段

とを含むオブジェクト結合プログラムが格納されてなることを特徴とするオブジェクト指向プログラム記憶媒体。

【請求項12】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、該メソッドIDにより特定されるメソッドが実行されるオブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

前記出力インストラクションバス部をもつ第1のオブジェクトが、メッセージを発行するにあたり、該メッセージに対応するメソッド要素リストに配列されたメソッド要素を参照して、該メソッド要素にポインタが格納されてなる第2のオブジェクトを、該メソッド要素に格納されてなるメソッドIDを引数として呼び出すものであることを特徴とする請求項11記載のオブジェクト指向プログラム記憶媒体。

【請求項13】 データと処理とを有する複数のオブジェクトが格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、該第1のオブジェクトのメッセージと該第2のオブジェクトのメソッドとを対応づけることにより、該第1のオブジェクトと該第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、該他オブジェクトのメッセージと自オブジェクトのメソッドとの対応をあらわす入力インストラクションタグテーブルを生成する入力インストラクションタグテーブル生成手段とを含むオブジェクト結合プログラムが格納されてなることを特徴とするオブジェクト指向プログラム記憶媒体。

【請求項14】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、該メソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタ

とを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、
前記入力インストラクションタグテーブル生成手段が、前記入力インストラクションタグテーブルを生成するとともに、該入力インストラクションタグテーブルを、該入力インストラクションタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることを特徴とする請求項 13 記載のオブジェクト指向プログラム記憶媒体。

【請求項 15】 前記入力インストラクションタグテーブルが追加されてなるメソッド要素を持つ第 1 のオブジェクトが、該メソッド要素により特定される第 2 のオブジェクトを呼び出すにあたり、該メソッド要素に格納されたメソッド ID と入力インストラクションタグテーブルとを引数として該第 2 のオブジェクトを呼び出すものであることを特徴とする請求項 14 記載のオブジェクト指向プログラム記憶媒体。

【請求項 16】 前記第 2 のオブジェクトが、前記第 1 のオブジェクトから該第 2 のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である入力インストラクションタグテーブルを参照し、該第 2 のオブジェクトのメッセージに対応する前記第 1 のオブジェクトのメソッドを実行するものであることを特徴とする請求項 15 記載のオブジェクト指向プログラム記憶媒体。

【請求項 17】 前記第 2 のオブジェクトが、前記第 1 のオブジェクトから該第 2 のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である入力インストラクションタグテーブルを参照し、該第 2 のオブジェクトのメッセージに対応する該第 2 のオブジェクトのメソッド要素リストに、該第 2 のオブジェクトのメッセージに対応づけられた前記第 1 のオブジェクトのメソッドに関するメソッド要素を追加するものであることを特徴とする請求項 15 記載のオブジェクト指向プログラム記憶媒体。

【請求項 18】 前記第 2 のオブジェクトが、第 3 のオブジェクトを生成する手段を有し、前記第 1 のオブジェクトから該第 2 のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である入力インストラクションタグテーブルを参照し、生成した第 3 のオブジェクトのメッセージに対応する該第 3 のオブジェクトのメソッド要素リストに、該第 3 のオブジェクトのメッセージに対応

づけられた前記第 1 のオブジェクトのメソッドに関するメソッド要素を追加するものであることを特徴とする請求項 1 5 記載のオブジェクト指向プログラム記憶媒体。

【請求項 1 9】 データと処理とを有する複数のオブジェクトが格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第 1 のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第 2 のオブジェクトとの間で、該第 1 のオブジェクトのメッセージと該第 2 のオブジェクトのメソッドとを対応づけることにより、該第 1 のオブジェクトと該第 2 のオブジェクトの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、該他オブジェクトのメソッドと自オブジェクトのメッセージとの対応をあらわす出力インストラクションタグテーブルを生成する出力インストラクションタグテーブル生成手段とを含むオブジェクト結合プログラムが格納されてなることを特徴とするオブジェクト指向プログラム記憶媒体。

【請求項 2 0】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッド ID と、該メソッド ID により特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

前記出力インストラクションタグテーブル生成手段が、前記出力インストラクションタグテーブルを生成するとともに、該出力インストラクションタグテーブルを、該出力インストラクションタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることを特徴とする請求項 1 9 記載のオブジェクト指向プログラム記憶媒体。

【請求項 2 1】 前記出力インストラクションタグテーブルが追加されてなるメソッド要素を持つ第 1 のオブジェクトが、該メソッド要素により特定される第 2 のオブジェクトを呼び出すにあたり、該メソッド要素に格納されたメソッド

IDと出力インストラクションタグテーブルとを引数として該第2のオブジェクトを呼び出すものであることを特徴とする請求項20記載のオブジェクト指向プログラム記憶媒体。

【請求項22】 前記第2のオブジェクトが、前記第1のオブジェクトから該第2のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である出力インストラクションタグテーブルを参照し、前記第1のオブジェクトのメッセージに対応する該第1のオブジェクトのメソッド要素リストに、該第1のオブジェクトのメッセージに対応づけられた該第2のオブジェクトのメソッドに関するメソッド要素を追加するものであることを特徴とする請求項21記載のオブジェクト指向プログラム記憶媒体。

【請求項23】 前記第2のオブジェクトが、第3のオブジェクトを生成する手段を有し、前記第1のオブジェクトから該第2のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である出力インストラクションタグテーブルを参照し、前記第1のオブジェクトのメッセージに対応する該第1のオブジェクトのメソッド要素リストに、該第1のオブジェクトのメッセージに対応づけられた前記第3のオブジェクトのメソッドに関するメソッド要素を追加するものであることを特徴とする請求項21記載のオブジェクト指向プログラム記憶媒体。

【請求項24】 データと処理とを有する複数のオブジェクトが格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、該第1のオブジェクトのメッセージと該第2のオブジェクトのメソッドとを対応づけることにより、該第1のオブジェクトと該第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、該他オブジェクトの、データが格納されるデータ格納領域へのポイ

ンタが配列されるデータ要素リストを特定するデータ要素リストIDと、自オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リストIDとの対応をあらわす入力データタグテーブルを生成する入力データタグテーブル生成手段とを含むオブジェクト結合プログラムが格納されてなることを特徴とするオブジェクト指向プログラム記憶媒体。

【請求項25】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、該メソッドIDにより特定されるメソッドが実行される他のオブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

前記入力データタグテーブル生成手段が、前記入力データタグテーブルを生成するとともに、該入力データタグテーブルを、該入力データタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることを特徴とする請求項24記載のオブジェクト指向プログラム記憶媒体。

【請求項26】 前記入力データタグテーブルが追加されてなるメソッド要素を持つ第1のオブジェクトが、該メソッド要素により特定される第2のオブジェクトを呼び出すにあたり、該メソッド要素に格納されたメソッドIDと入力データタグテーブルとを引数として該第2のオブジェクトを呼び出すものであることを特徴とする請求項25記載のオブジェクト指向プログラム記憶媒体。

【請求項27】 前記第2のオブジェクトが、前記第1のオブジェクトから該第2のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である入力データタグテーブルを参照し前記第1のオブジェクトのポインタ要素リストIDを取得して、該第1のオブジェクトの、該ポインタ要素リストIDにより特定されるポインタ要素リストを生成するとともに、該第2のオブジェクトの、該ポインタ要素リストIDに対応するデータ要素リストIDにより特定されるデータ要素リストを生成し、前記第1のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、該第2のオブジェクトのデータ要素リストに配列されたポインタを書き込むものであることを特徴とする請求項

2 6 記載のオブジェクト指向プログラム記憶媒体。

【請求項 2 8】 前記第 2 のオブジェクトが、第 3 のオブジェクトを生成する手段を有し、前記第 1 のオブジェクトから該第 2 のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である入力データタグテーブルを参照し前記第 1 のオブジェクトのポインタ要素リスト I D を取得して、該第 1 のオブジェクトの、該ポインタ要素リスト I D により特定されるポインタ要素リストを生成するとともに、前記第 3 のオブジェクトの、該ポインタ要素リスト I D に対応するデータ要素リスト I D により特定されるデータ要素リストを生成し、前記第 1 のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、該第 3 のオブジェクトのデータ要素リストに配列されたポインタを書き込むものであることを特徴とする請求項 2 6 記載のオブジェクト指向プログラム記憶媒体。

【請求項 2 9】 データと処理とを有する複数のオブジェクトが格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第 1 のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、該メッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第 2 のオブジェクトとの間で、該第 1 のオブジェクトのメッセージと該第 2 のオブジェクトのメソッドとを対応づけることにより、該第 1 のオブジェクトと該第 2 のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、該他オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リスト I D と、自オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リスト I D との対応をあらわす出力データタグテーブルを生成する出力データタグテーブル生成手段とを含むオブジェクト結合プログラムが格納されてなることを特徴とするオブジェクト指向プログラム記憶媒体。

【請求項30】 前記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、該メソッドIDにより特定されるメソッドが実行される他のオブジェクトへのポインタを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

前記出力データタグテーブル生成手段が、前記出力データタグテーブルを生成するとともに、該出力データタグテーブルを、該出力データタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることを特徴とする請求項29記載のオブジェクト指向プログラム記憶媒体。

【請求項31】 前記出力データタグテーブルが追加されてなるメソッド要素を持つ第1のオブジェクトが、該メソッド要素により特定される第2のオブジェクトを呼び出すにあたり、該メソッド要素に格納されたメソッドIDと出力データタグテーブルとを引数として該第2のオブジェクトを呼び出すものであることを特徴とする請求項30記載のオブジェクト指向プログラム記憶媒体。

【請求項32】 前記第2のオブジェクトが、前記第1のオブジェクトから該第2のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である出力データタグテーブルを参照し前記第1のオブジェクトのデータ要素リストIDを取得して、該第1のオブジェクトの、該データ要素リストIDにより特定されるデータ要素リストを生成するとともに、該第2のオブジェクトの、該データ要素リストIDに対応するポインタ要素リストIDにより特定されるポインタ要素リストを生成し、該第2のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、前記第1のオブジェクトのデータ要素リストに配列されたポインタを書き込むものであることを特徴とする請求項31記載のオブジェクト指向プログラム記憶媒体。

【請求項33】 前記第2のオブジェクトが、第3のオブジェクトを生成する手段を有し、前記第1のオブジェクトから該第2のオブジェクトに向けたメッセージを受け付け、該メッセージの引数である出力データタグテーブルを参照し前記第1のオブジェクトのデータ要素リストIDを取得して、該第1のオブジェクトの、該データ要素リストIDにより特定されるデータ要素リストを生成する

とともに、前記第3のオブジェクトの、該データ要素リストIDに対応するポインタ要素リストIDにより特定されるポインタ要素リストを生成し、該第3のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、前記第1のオブジェクトのデータ要素リストに配列されたポインタを書き込むものであることを特徴とする請求項31記載のオブジェクト指向プログラム記憶媒体。

【請求項34】 前記第2のオブジェクトが、前記第1のオブジェクトから該第2のオブジェクトに向けたメッセージを受け付け、該メッセージの引数であるメソッドIDにより特定されるメソッドを実行するものであることを特徴とする請求項12、15、21、26、又は31記載のオブジェクト指向プログラム記憶媒体。

【請求項35】 電子化された画像情報に基づく画像を表示画面上に表示する画像表示装置の該表示画面上に、オブジェクト指向プログラミングにより生成される複数のオブジェクトと、これら複数のオブジェクト間のデータおよび制御の流れを示す配線とを表示するオブジェクト間ネットワーク表示方法において、

前記表示画面が複数の升目に区切られてなる升目1つからなる表示領域内、もしくは隣接する複数の升目が連結されてなる表示領域それぞれに、1つのオブジェクトを表示するオブジェクト表示部分領域と、複数のオブジェクト間を結ぶ配線を表示する配線表示部分領域とを、隣接する2つの表示領域それぞれのオブジェクト表示部分領域どうしの間に配線表示部分領域が形成されるように定め、

前記表示画面上に、複数のオブジェクトそれぞれが、各表示領域内のオブジェクト表示部分領域に1つずつ配置されるとともに、これら複数のオブジェクトどうしを結ぶ配線が、複数の表示領域に跨って連なる配線表示部分領域に表示されてなる第1の画像を表示することを特徴とするオブジェクト間ネットワーク表示方法。

【請求項36】 前記第1の画像を構成する複数のオブジェクトのうちの所定のオブジェクトが、該所定のオブジェクトよりも下層側の複数のオブジェクトとこれら複数のオブジェクト間を結ぶ配線とから成るサブネットワークで構成されるものであって、

前記第 1 の画像に代えて、該第 1 の画像内の前記所定のオブジェクトの表示に代わり該所定のオブジェクトのサブネットワークが表示されてなる第 2 の画像を表示するにあたり、前記第 1 の画像内の、前記所定のオブジェクトの表示領域よりも拡大された表示領域に該サブネットワークが表示されるとともに、該サブネットワークの表示領域の上下および左右に並ぶ表示領域がそれぞれ左右および上下に拡大された表示領域に変更され、さらに、該サブネットワークの表示領域に対し斜めに位置する表示領域については前記第 1 の画像と同一の寸法で表示されてなる第 2 の画像を表示することを特徴とする請求項 3 5 記載のオブジェクト間ネットワーク表示方法。

【請求項 3 7】 前記第 1 の画像を構成する複数のオブジェクトのうちの所定のオブジェクトが、該所定のオブジェクトよりも下層側の複数のオブジェクトとこれら複数のオブジェクト間を結ぶ配線とから成るサブネットワークで構成されるものであって、

前記第 1 の画像に代えて、該第 1 の画像内の前記所定のオブジェクトの表示に代わり該所定のオブジェクトのサブネットワークが表示されてなる第 2 の画像を表示するにあたり、前記第 1 の画像内の、前記所定のオブジェクトの表示領域よりも拡大された表示領域に該サブネットワークが表示されるとともに、該第 2 の画像の周縁に位置する表示領域、および該表示領域の、該第 2 の画像の外周に接する辺の位置および寸法が、前記第 1 の画像の周縁に位置する表示領域、および該表示領域の、該第 1 の画像の外周に接する辺の位置および寸法と比べ、同一のオブジェクトが表示された表示領域であり、かつ外周に接する辺の位置および寸法も同一となるように、前記サブネットワークの表示領域を除く表示領域が前記第 1 の画像上の対応する表示領域と比べ変形してなる第 2 の画像を表示することを特徴とする請求項 3 5 記載のオブジェクト間ネットワーク表示方法。

【請求項 3 8】 前記第 1 の画像を表示するにあたり、前記表示領域内のオブジェクト表示部分領域の形状および寸法が、該表示領域の形状および寸法に同じ形状および寸法に規格化されてなることを特徴とする請求項 3 5 記載のオブジェクト間ネットワーク表示方法。

【請求項 3 9】 前記第 1 の画像を表示するにあたり、先ず複数のオブジェ

クトを表示し、次いで、これら複数のオブジェクト間を、データもしくは制御の流れの方向が所定長部分毎に繰り返し表示されてなる配線で連結している状態を表示することを特徴とする請求項 3 5 記載のオブジェクト間ネットワーク表示方法。

【請求項 4 0】 前記第 1 の画像を表示するにあたり、中心線と、該中心線の両脇に沿う、該中心線とは異なる表示態様の縁線とからなる配線であって、相互に交差する配線のうち、同一のデータもしくは制御の流れを表わす配線については中心線どうしが連続し、異なるデータもしくは制御の流れを表わす配線については一方の配線の中心線を他方の配線の縁線に接する位置もしくは近接する位置で分断してなる配線を表示することを特徴とする請求項 3 5 記載のオブジェクト間ネットワーク表示方法。

【請求項 4 1】 データと処理とを有するオブジェクト複数を、指示に応じて相互に結合するオブジェクト指向プログラミング支援装置において、

オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジェクトであって 1 つのオブジェクトの内部に 1 つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する表示手段と、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、

オブジェクトの階層構造を構築する階層構造構築手段と、

前記オブジェクト結合手段に対しオブジェクト間を結ぶ配線を指示入力するとともに、前記階層構造構築手段に対し、オブジェクトの、階層構造上の配置位置を指示入力する操作子とを備え、

前記階層構造構築手段が、前記操作子からの指示に応じて、指定されたオブジ

ェクトである本体オブジェクトを複写して分身オブジェクトを作成し、該分身オブジェクトを、該本体オブジェクトが配置された階層とは異なる階層に配置する手段を有し、

前記オブジェクト結合手段が、前記分身オブジェクトが配置された階層における配線において、前記操作子からの、前記分身オブジェクトと他オブジェクトとの間の配線の指示を受け付けるとともに、該分身オブジェクトと該分身オブジェクトに対応する本体オブジェクトとを1つのオブジェクトとした結合構造を構築するものであることを特徴とするオブジェクト指向プログラミング支援装置。

【請求項42】 データと処理とを有するオブジェクト複数を、指示に応じて相互に結合するオブジェクト指向プログラミング支援装置において、

オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジェクトであって1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する表示手段と、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、

オブジェクトの階層構造を構築する階層構造構築手段と、

前記オブジェクト結合手段に対しオブジェクト間を結ぶ配線を指示入力するとともに、前記階層構造構築手段に対し、オブジェクトの、階層構造上の配置位置を指示入力する操作子とを備え、

前記オブジェクト結合手段が、前記操作子からの、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトの、他オブジェクトとの結合構造を解除して、置き換え前のオブジェクトの他オブジェクトとの結合構造を、置き換え後のオブジェクトに継承させるものであり、

前記階層構造構築手段が、前記操作子からの、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトが配置されていた階層に、該置き換え前のオブジェクトに代えて、置き換え後のオブジェクトを配置するものであることを特徴とするオブジェクト指向プログラミング支援装置。

【請求項43】 データと処理とを有するオブジェクト複数を、指示に応じて相互に結合するオブジェクト指向プログラミング支援装置において、

オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジェクトであって1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する表示手段と、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、

オブジェクトの階層構造を構築する階層構造構築手段と、

前記オブジェクト結合手段に対しオブジェクト間を結ぶ配線を指示入力するとともに、前記階層構造構築手段に対し、オブジェクトの、階層構造上の配置位置を指示入力する操作子とを備え、

前記階層構造構築手段が、前記操作子からの、所定の階層に配置されたオブジェクトの中から複数のオブジェクトを指定して該複数のオブジェクトを一段下の階層に再配置する旨の指示に応じて、該複数のオブジェクト相互間の結合構造、および該複数のオブジェクトと、該複数のオブジェクト以外の他オブジェクトとの間の結合構造が保持されるように、該複数のオブジェクトを該一段下の階層に再配置するとともに該所定の階層に該複数のオブジェクトを内包するオブジェクトを生成して配置するものであることを特徴とするオブジェクト指向プログラミング支援装置。

【請求項44】 データと処理とを有するオブジェクト複数を、指示に応じて相互に結合するオブジェクト指向プログラミング支援装置において、

オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジェクトであって1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する表示手段と、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、

オブジェクトの階層構造を構築する階層構造構築手段と、

前記オブジェクト結合手段に対しオブジェクト間を結ぶ配線を指示入力するとともに、前記階層構造構築手段に対し、オブジェクトの、階層構造上の配置位置を指示入力する操作子とを備え、

前記表示手段が、前記操作子による指示に応じて、指定された1つのメッセージ端子に接続されたメソッド端子が複数存在する場合の、これら複数のメソッド端子に対応する複数のメソッドの実行順序をあらわすリストを表示する手段を有し、

前記オブジェクト結合手段が、前記操作子による指示に応じて、前記表示手段に表示されたリストにあらわれた複数のメソッドの実行順序が変更された結合構造を再構築する手段を有するものであることを特徴とするオブジェクト指向プログラミング支援装置。

【請求項45】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体において、

前記オブジェクトが、オブジェクト本体をあらわすブロックと、自オブジェク

ト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されるとともに、1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されるオブジェクトであって、複数のオブジェクトの端子間を配線で結ぶ指示がなされるものであり、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、オブジェクトの階層構造を構築する階層構造構築手段とを含むプログラムであって、

前記階層構造構築手段が、指定されたオブジェクトである本体オブジェクトを複製して分身オブジェクトを作成し、該分身オブジェクトを、該本体オブジェクトが配置された階層とは異なる階層に配置する手段を有し、前記オブジェクト結合手段が、前記分身オブジェクトが配置された階層における配線において、前記分身オブジェクトと他オブジェクトとの間の配線の指示を受け付けるとともに、該分身オブジェクトと該分身オブジェクトに対応する本体オブジェクトとを1つのオブジェクトとした結合構造を構築するものであるプログラムが格納されてなることを特徴とするオブジェクト指向プログラミング用プログラム記憶媒体。

【請求項46】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体において、

前記オブジェクトが、オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されるとともに、1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されるオブジェクトであって、複数の

オブジェクトの端子間を配線で結ぶ指示がなされるものであり、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、オブジェクトの階層構造を構築する階層構造構築手段とを含むプログラムであって、

前記オブジェクト結合手段が、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトの、他オブジェクトとの結合構造を解除して、置き換え前のオブジェクトの他オブジェクトとの結合構造を、置き換え後のオブジェクトに継承させるものであり、前記階層構造構築手段が、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトが配置されていた階層に、該置き換え前のオブジェクトに代えて、置き換え後のオブジェクトを配置するものであるプログラムが格納されてなることを特徴とするオブジェクト指向プログラミング用プログラム記憶媒体。

【請求項 4 7】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体において、

前記オブジェクトが、オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されるとともに、1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されるオブジェクトであって、複数のオブジェクトの端子間を配線で結ぶ指示がなされるものであり、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、オブジェクトの階層構造を構築する階層構造構築手段とを含むプログラムであって、

前記階層構造構築手段が、所定の階層に配置されたオブジェクトの中から指定

された複数のオブジェクトを一段下の階層に再配置する旨の指示に応じて、該複数のオブジェクト相互間の結合構造、および該複数のオブジェクトと、該複数のオブジェクト以外の他オブジェクトとの間の結合構造が保持されるように、該複数のオブジェクトを該一段下の階層に再配置するとともに該所定の階層に該複数のオブジェクトを内包するオブジェクトを生成して配置するものであるプログラムが格納されてなることを特徴とするオブジェクト指向プログラミング用プログラム記憶媒体。

【請求項48】 データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体において、

前記オブジェクトが、オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されるとともに、1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されるオブジェクトであって、複数のオブジェクトの端子間を配線で結ぶ指示がなされるものであり、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、該配線に対応した結合構造を構築するオブジェクト結合手段と、オブジェクトの階層構造を構築する階層構造構築手段とを含むプログラムであって、

前記オブジェクト結合手段が、指定された1つのメッセージ端子に接続されたメソッド端子が複数存在する場合の、これら複数のメソッド端子に対応する複数のメソッドの実行順序をあらわすリストを作成する手段と、該リストにあらわれた複数のメソッドの実行順序の変更指示に応じて該複数のメソッドの実行順序が変更された結合構造を再構築する手段とを有するものであるプログラムが格納されてなることを特徴とするオブジェクト指向プログラミング用プログラム記憶媒体。

【請求項49】 他オブジェクトで発行されたメッセージにより発火して所

定の既存ソフトウェアのイベントを発行するメソッドを含む、該既存ソフトウェアと合わせて1つのオブジェクトとして作用するコンポーネントが格納されてなることを特徴とするコンポーネント記憶媒体。

【請求項 5 0】 前記コンポーネントが、前記メソッドを含むとともに該メソッドの実行により前記イベントが発行されたことを他オブジェクトに向けて通知するメッセージを含むものであることを特徴とする請求項 4 9 記載のコンポーネント記憶媒体。

【請求項 5 1】 所定の既存ソフトウェアの所定のイベントの発生を受けて該所定のイベントが発生したことを他オブジェクトに向けて通知するメッセージを含む、該既存ソフトウェアと合わせて1つのオブジェクトとして作用するコンポーネントが格納されてなることを特徴とするコンポーネント記憶媒体。

【請求項 5 2】 メソッドの作成およびメッセージの作成を選択的に指示入力する第 1 の操作子と、

所定の既存ソフトウェアの所望のイベントの発行を指示入力する第 2 の操作子と、

前記既存ソフトウェアと合わせて1つのオブジェクトとして作用するコンポーネントを作成するコンポーネント作成手段であって、前記第 1 の操作子の操作によりメソッドの作成が指示され次いで前記第 2 の操作子の操作により前記既存ソフトウェアの所定のイベントが発行された場合に、該コンポーネントに、他オブジェクトで発行されたメッセージにより発火して該所定のイベントを発行するメソッドを作成し、前記第 1 の操作子によりメッセージの作成が指示され次いで前記第 2 の操作子により前記既存ソフトウェアの所定のイベントの発行が指示された場合に、該コンポーネントに、該所定のイベントの発生に応答して該所定のイベントが発生したことを他オブジェクトに通知するためのメッセージを作成するコンポーネント作成手段とを備えたことを特徴とするコンポーネント作成装置。

【請求項 5 3】 他オブジェクトで発行されたメッセージにより発火して所定の既存ソフトウェアのイベントを発行するメソッドと、該メソッドの実行により前記イベントが発行されたことを他オブジェクトに向けて通知するメッセージとを含み該所定の既存ソフトウェアと合わせて1つのオブジェクトとして作用す

るコンポーネントを、1つもしくは複数の既存ソフトウェアに関して格納するコンポーネントファイルと、

既存ソフトウェアについてイベントの発行を指示入力するための操作子と、

前記操作子の操作に応じて順次発行された、1つもしくは複数の既存ソフトウェアに関するイベントのリストを格納するイベントログファイルと、

該イベントログファイルから順次にイベントを取り出して、前回取り出したイベントと同一のイベントが発行されたことを他オブジェクトに向けて通知するメッセージを含むコンポーネントの該メッセージと、今回取り出したイベントと同一のイベントを発行するメソッドを含むコンポーネントの該メソッドとを結合するコンポーネント結合手段とを備えたことを特徴とするオブジェクト指向プログラミング支援装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、オブジェクト指向プログラミングを行なうオブジェクト指向プログラミング装置、オブジェクト指向プログラミングを支援するオブジェクト指向プログラミング支援装置、オブジェクトの一部をなすコンポーネントを作成するコンポーネント作成装置、オブジェクト指向プログラムが格納されたオブジェクト指向プログラム記憶媒体、オブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体、コンポーネントが格納されたコンポーネント記憶媒体、および、オブジェクト指向プログラミングにより生成される複数のオブジェクトについて、データの共有によるデータ結合や、オブジェクト間の制御フローの結合などをオブジェクトのネットワークとして視覚的に表示するオブジェクト間ネットワーク表示方法に関する。

【0002】

【従来の技術】

コンピュータに組み込まれて動作するプログラムを記述するにあたり、従前には、関数名（コマンド）と変数を順番に記述してプログラミングを行なっていた。このようなプログラミング方法の場合、全てをコマンドで記述していく必要が

あるため、コマンドをいちいちマニュアルで調べたり、あるいは多数のコマンドを覚えている必要がある。このようなコマンドはプログラム言語ごとに異なり、あるプログラム言語のコマンドを多数覚えても、他のプログラム言語で記述しようとするときは、そのプログラム言語のコマンドを覚え直さなければならないという不都合もある。さらには、プログラムのフォーマットもプログラム言語毎に異なっている。これらのことが、プログラムの記述を難かしくしており、プログラムの開発は、素人には中々入りこめない特殊な分野であるかのような印象を与えていた。近年、プログラムがますます大規模化、複雑化してきており、プログラムの開発を容易にし、また、一旦開発したプログラムの再利用を図る必要性がますます高まってきている。

【0003】

このような背景の中で、近年、オブジェクト指向プログラミングが広く採用されてきている。ここで、オブジェクトとは、「データ」と「そのデータに関する処理」とを併せ持ったデータ構造をいい、オブジェクト指向とは、「データ」と「そのデータに関する処理」、すなわちオブジェクトを単位として取り扱う考え方をいう。このオブジェクト指向プログラミングにおいても、もともとは個々のソフトウェア（オブジェクト）を作成する必要があるか、個々のオブジェクトを一旦作成した後は、あるオブジェクトから別のオブジェクトを呼び出すという、オブジェクトどうしの接続関係を記述することによりプログラミングが完結することになる。このような考え方を取り入れたオブジェクト指向プログラミングにより、大規模化、複雑化するソフトウェアの使い勝手、作り方、メンテナンスが格段に向上すると期待されている。

【0004】

オブジェクト指向プログラムにおいては、あるオブジェクトから別のオブジェクトを呼び出す処理は、呼び出す側のオブジェクトから呼び出される側のオブジェクトに向けてメッセージを発行し、呼び出される側のオブジェクトではそのメッセージを受け取ってそのメッセージに対応するメソッド（処理）を実行するという、メッセージ、メソッドという概念が用いられる。メソッドの実行に必要なデータは、従来は、メッセージの引数として与えられている。

【 0 0 0 5 】**【 発明が解決しようとする課題 】**

オブジェクト指向プログラミングが目指す目標の1つとして、一度作成したソフトウェア（オブジェクト）を、システムが変更されても再利用できるようにすることにある。これを行なうには、比較的小さく単純なオブジェクトを作成する必要がある。

【 0 0 0 6 】

ところが、従来は、受け取ったメッセージにどのメソッドが対応するかを認識するのに時間がかかったり、メッセージを発行するオブジェクトからメソッドを実行するオブジェクトへのデータの受け渡しに時間がかかり、一般に、オブジェクト指向プログラムは実行速度が遅いと言われている。

この実行速度を向上させるために、従来、1つのオブジェクト内での処理を増やし、他のオブジェクトに向けてメッセージを発行する機会を少なくする手法が採られており、この場合、1つのオブジェクト内での処理が複雑化し、オブジェクトが大型化し、上述の再利用性と相反することとなり、オブジェクト指向プログラミングにおいてソフトウェアの再利用性が進まない原因の1つとなっている。

【 0 0 0 7 】

また、オブジェクト指向プログラミングを推進するにあたって大きな問題となるのは、オブジェクト指向という考えには基づいていない、従来から蓄積されてきた多数のソフトウェアの取扱いである。これまでに提案されているオブジェクト指向プログラミング手法では、既存のソフトウェアの再利用性がきわめて低いという問題もある。

【 0 0 0 8 】

本発明は、上記事情に鑑み、複数のオブジェクトを、それら複数のオブジェクト間で情報が効率的に流れるように相互に結合する機能を備えたオブジェクト指向プログラミング装置、複数のオブジェクトと、それら複数のオブジェクトをそれら複数のオブジェクト間で情報が効率的に流れるように相互に結合するためのオブジェクト結合プログラムとが格納されたオブジェクト指向プログラム記憶媒

体、オブジェクトどうしの結合関係を定義するオブジェクト指向プログラミングの容易化が図られたオブジェクト指向プログラミング支援装置、オブジェクト指向プログラミングを容易に行なうためのプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体、既存のソフトウェアをオブジェクトとして取り扱うことができるように、その既存のソフトウェアと合わせて1つのオブジェクトとして作用するコンポーネントを作成する機能を備えたコンポーネント作成装置、そのようなコンポーネントが格納されたコンポーネント記憶媒体、および、オブジェクトどうしの結合関係を定義するオブジェクト結合プログラミングを行なうのに好適な、オブジェクトとオブジェクトどうしの結合関係とからなるオブジェクト間ネットワークの表示方法を提供することを目的とする。

【0 0 0 9】

【課題を解決するための手段】

上記目的を達成する本発明のオブジェクト指向プログラミング装置のうちの第1のオブジェクト指向プログラミング装置は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されてなるデータ要素リストを生成するデータ要素リスト生成手段と、

オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されてなるポインタ要素リストを生成するポインタ要素リスト生成手段と、

データ要素リストを持つ第3のオブジェクトとポインタ要素リストを持つ第4のオブジェクトとの間で、第4のオブジェクトのポインタ要素リストに配列され

たポインタが指し示すポインタ格納領域に、第3のオブジェクトのデータ要素リストに配列されたポインタを書き込むことにより、第3のオブジェクトと第4のオブジェクトとの間でのデータの授受を可能ならしめるデータ結合手段とを備えたことを特徴とする。

【0010】

ここで、本発明の第1のオブジェクト指向プログラミング装置において、上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであることが好ましい。

【0011】

また、上記目的を達成する本発明のオブジェクト指向プログラミング装置のうちの第2のオブジェクト指向プログラミング装置は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトのメッセージと自オブジェクトのメソッドとの対応をあらわす入力インストラクションタグテーブルを生成する入力インストラクションタグテーブル生成手段とを備えたことを特徴とする。

【0012】

ここで、本発明の第2のオブジェクト指向プログラミング装置において、上記インストラクション結合手段が、自オブジェクトのメッセージに対応する

他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

上記入力インストラクションタグテーブル生成手段が、入力インストラクションタグテーブルを生成するとともに、その入力インストラクションタグテーブルを、その入力インストラクションタグテーブルに対応する他オブジェクトへの入インタを含むメソッド要素に追加するものであることが好ましい。

【0013】

ここで、入力インストラクションタグテーブルをメソッド要素に追加する態様の1つとして、そのメソッド要素には、直接的には、その入力インストラクションタグテーブルへのポインタを書き込んでもよい。

【0014】

【課題を解決するための手段】

上記目的を達成する本発明のオブジェクト指向プログラミング装置のうちの第1のオブジェクト指向プログラミング装置は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されてなるデータ要素リストを生成するデータ要素リスト生成手段と、

オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されてなるポインタ要素リストを生成するポインタ要素リスト生成手段と、

データ要素リストを持つ第3のオブジェクトとポインタ要素リストを持つ第4

のオブジェクトとの間で、第 4 のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、第 3 のオブジェクトのデータ要素リストに配列されたポインタを書き込むことにより、第 3 のオブジェクトと第 4 のオブジェクトとの間でのデータの授受を可能ならしめるデータ結合手段とを備えたことを特徴とする。

【0 0 1 5】

ここで、本発明の第 1 のオブジェクト指向プログラミング装置において、上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッド ID と、そのメソッド ID により特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであることが好ましい。

【0 0 1 6】

また、上記目的を達成する本発明のオブジェクト指向プログラミング装置のうちの第 2 のオブジェクト指向プログラミング装置は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第 1 のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第 2 のオブジェクトとの間で、第 1 のオブジェクトのメッセージと第 2 のオブジェクトのメソッドとを対応づけることにより、第 1 のオブジェクトと第 2 のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトのメッセージと自オブジェクトのメソッドとの対応をあらわす入力インストラクションタグテーブルを生成する入力インストラクションタグテーブル生成手段とを備えたことを特徴とする。

【0 0 1 7】

ここで、本発明の第 2 のオブジェクト指向プログラミング装置において、

上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

上記入力インストラクションタグテーブル生成手段が、入力インストラクションタグテーブルを生成するとともに、その入力インストラクションタグテーブルを、その入力インストラクションタグテーブルに対応する他オブジェクトへの入インタを含むメソッド要素に追加するものであることが好ましい。

【0018】

ここで、入力インストラクションタグテーブルをメソッド要素に追加する態様の1つとして、そのメソッド要素には、直接的には、その入力インストラクションタグテーブルへのポインタを書き込んでもよい。

また、上記目的を達成する本発明のオブジェクト指向プログラミング装置のうちの第3のオブジェクト指向プログラミング装置は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトのメソッドと自オブジェクトのメッセージとの対応をあらわす出力インストラクションタグテーブルを生成する出力インストラクションタグテーブル生成手段とを備えたことを特徴とする。

【0019】

ここで、本発明の第3のオブジェクト指向プログラミング装置においても、

上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

上記出力インストラクションタグテーブル生成手段が、出力インストラクションタグテーブルを生成するとともに、その出力インストラクションタグテーブルを、その出力インストラクションタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることが好ましい。

【0020】

ここで、出力インストラクションタグテーブルをメソッド要素に追加する態様の1つとして、そのメソッド要素には、直接的には、その出力インストラクションタグテーブルへのポインタを書き込んでもよい。

さらに、上記目的を達成する本発明の第4のオブジェクト指向プログラミング装置は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リストIDと、自オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リストIDとの対応をあらわす入力データタグテーブルを生成する入力データタグテーブル生成手段とを備えたことを特徴とする。

【0021】

ここで、上記本発明の第4のオブジェクト指向プログラミング装置において、
上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

上記入力データタグテーブル生成手段が、入力データタグテーブルを生成するとともに、その入力データタグテーブルを、その入力データタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることが好ましい。

【0022】

ここで、入力データタグテーブルをメソッド要素に追加する態様の1つとして、そのメソッド要素には、直接的には、その入力データタグテーブルへのポインタを書き込んでもよい。

さらに、上記目的を達成する本発明の第5のオブジェクト指向プログラミング装置は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミング装置において、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リストIDと、自オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リストIDとの対応をあら

わす出力データタグテーブルを生成する出力データタグテーブル生成手段とを備えたことを特徴とする。

【0023】

ここで、上記本発明の第5のオブジェクト指向プログラミング装置においても、

上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される第2のオブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、上記出力データタグテーブル生成手段が、出力データタグテーブルを生成するとともに、その出力データタグテーブルを、その出力データタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることが好ましい。

【0024】

ここで、出力データタグテーブルをメソッド要素に追加する態様の1つとして、そのメソッド要素には、直接的には、その出力データタグテーブルへのポインタを書き込んでもよい。

また、上記目的を達成する本発明のオブジェクト指向プログラム記憶媒体のうちの、第1のオブジェクト指向プログラム記憶媒体は、

データと処理とを有するの複数オブジェクトが格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されてなるデータ要素リストを生成するデータ要素リスト生成手段と、

オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポイ

ンタが配列されてなるポインタ要素リストを生成するポインタ要素リスト生成手段と、

データ要素リストを持つ第3のオブジェクトとポインタ要素リストを持つ第4のオブジェクトとの間で、第4のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、第3のオブジェクトのデータ要素リストに配列されたポインタを書き込むことにより、第3のオブジェクトと第4のオブジェクトとの間でのデータの授受を可能ならしめるデータ結合手段とを含むオブジェクト結合プログラムが格納されてなることを特徴とする。

【0025】

ここで、上記第1のオブジェクト指向プログラム記憶媒体において、

上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行されるオブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

出力インストラクションバス部を持つ第1のオブジェクトが、メッセージ発行するにあたり、そのメッセージに対応するメソッド要素リストに配列されたメソッド要素を参照して、そのメソッド要素にポインタが格納されてなる第2のオブジェクトを、そのメソッド要素に格納されてなるメソッドIDを引数として呼び出すものであることが好ましい。

【0026】

その場合に、上記第2のオブジェクトでは、第1のオブジェクトから第2のオブジェクトに向けたメッセージが受け付けられ、そのメッセージの引数であるメソッドIDにより特定されるメソッドが実行される。

また、上記目的を達成する本発明のオブジェクト指向プログラム記憶媒体のうちの第2のオブジェクト指向プログラム記憶媒体は、

データと処理とを有する複数のオブジェクトが格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応

するメソッドを起動する入力インストラクションバス部を持つ第 2 のオブジェクトとの間で、第 1 のオブジェクトのメッセージと第 2 のオブジェクトのメソッドとを対応づけることにより、第 1 のオブジェクトと第 2 のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトのメッセージと自オブジェクトのメソッドとの対応をあらわす入力インストラクションタグテーブルを生成する入力インストラクションタグテーブル生成手段とを含むオブジェクト結合プログラムが格納されてなることを特徴とする。

【0 0 2 7】

ここで、上記第 2 のオブジェクト指向プログラム記憶媒体において、

上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッド ID と、そのメソッド ID により特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

上記入力インストラクションタグテーブル生成手段が、入力インストラクションタグテーブルを生成するとともに、その入力インストラクションタグテーブルを、その入力インストラクションタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることが好ましい。

【0 0 2 8】

ここで、入力インストラクションタグテーブルをメソッド要素に追加する態様の 1 つとして、そのメソッド要素には、直接的には、その入力インストラクションタグテーブルへのポインタを書き込んでもよい。

入力インストラクションタグテーブルが追加されてなるメソッド要素を持つ第 1 のオブジェクトは、そのメソッド要素により特定される第 2 のオブジェクトを呼び出すにあたり、そのメソッド要素に格納されたメソッド ID と入力インストラクションタグテーブルとを引数として第 2 のオブジェクトを呼び出すものであってもよい。

【0 0 2 9】

ここで、入力インストラクションタグテーブルを引数として第2のオブジェクトを呼び出す態様の1つとして、直接的には、その入力インストラクションタグテーブルへのポインタを引数として第2のオブジェクトを呼び出してよい。

このようにして、第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行されると、第2のオブジェクトでは、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数であるメソッドIDにより特定されるメソッドが実行される。

ここで、第2のオブジェクトは、上記のようにして第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行された際、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数である入力インストラクションタグテーブルを参照し、第2のオブジェクトのメッセージに対応する第1のオブジェクトのメソッドを実行するものであってもよい。

【0030】

あるいは、第2のオブジェクトが、上記のようにして第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行された際に、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数である入力インストラクションタグテーブルを参照し、第2のオブジェクトのメッセージに対応する第2のオブジェクトのメソッド要素リストに、その第2のオブジェクトのメッセージに対応づけられた第1のオブジェクトのメソッドに関するメソッド要素を追加するものであることも好ましい形態である。

【0031】

さらには、第2のオブジェクトが、第3のオブジェクトを生成する手段と、を有し、上記のようにして第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行された際に、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数である入力インストラクションタグテーブルを参照し、生成した第3のオブジェクトのメッセージに対応する、その第3のオブジェクトのメソッド要素リストに、その第3のオブジェクトのメッセージに対応づけられた第1のオブジェクトのメソッドに関するメソッド要素を追加するものであることも好ましい形態である。

【0032】

ここで、第2のオブジェクトが第3のオブジェクトを生成するタイミングは、本発明においては特に限定されるものではなく、上記メッセージが発行された際に第3のオブジェクトを生成してもよく、あるいは、前もって生成しておいてもよい。

また、本発明のオブジェクト指向プログラム記憶媒体のうちの第3のオブジェクト指向プログラム記憶媒体は、

データと処理とを有する複数のオブジェクト、が格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトのメソッドと自オブジェクトのメッセージとの対応をあらわす出力インストラクションタグテーブルを生成する出力インストラクションタグテーブル生成手段とを含むオブジェクト結合プログラムが格納されてなることを特徴とする。

【0033】

ここで、上記第3のオブジェクト指向プログラム記憶媒体において、

上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

上記出力インストラクションタグテーブル生成手段が、出力インストラクションタグテーブルを生成するとともに、その出力インストラクションタグテーブルを、その出力インストラクションタグテーブルに対応する他オブジェクトへのポ

インタを含むメソッド要素に追加するものであることが好ましい。

【0034】

ここで、出力インストラクシオンタグテーブルをメソッド要素に追加する態様の1つとして、そのメソッド要素には、直接的には、その出力インストラクシオンタグテーブルへのポインタを書き込んでもよい。

出力インストラクシオンタグテーブルが追加されてなるメソッド要素を持つ第1のオブジェクトは、そのメソッド要素により特定される第2のオブジェクトを呼び出すにあたり、メソッド要素に格納されたメソッドIDと出力インストラクシオンタグテーブルとを引数として第2のオブジェクトを呼び出すものであってもよい。

【0035】

ここで、出力インストラクシオンタグテーブルを引数として第2のオブジェクトを呼び出す態様の1つとして、直接的には、その出力インストラクシオンタグテーブルへのポインタを引数として第2のオブジェクトを呼び出してもよい。

このようにして、第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行されると、第2のオブジェクトでは、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数であるメソッドIDにより特定されるメソッドが実行される。

【0036】

ここで、第2のオブジェクトは、上記のようにして第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行された際、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数である出力インストラクシオンタグテーブルを参照し、第1のオブジェクトのメッセージに対応する第1のオブジェクトのメソッド要素リストに、その第1のオブジェクトのメッセージに対応づけられた第2のオブジェクトのメソッドに関するメソッド要素を追加するものであってもよい。

【0037】

あるいは、第2のオブジェクトが、第3のオブジェクトを生成する手段と、を有し、上記のようにして第1のオブジェクトから第2のオブジェクトに向けてメ

ッセージが発行された際に、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数である出力インストラクションタグテーブルを参照し、第1のオブジェクトのメッセージに対応する第1のオブジェクトのメソッド要素リストに、その第1のオブジェクトのメッセージに対応づけられた第3のオブジェクトのメソッドに関するメソッドを追加するものであることも好ましい形態である。

【0038】

ここで、この第3のオブジェクト指向プログラム記憶媒体においても、前述した第2のオブジェクト指向プログラム記憶媒体の場合と同様、第2のオブジェクトが第3のオブジェクトを生成するタイミングは特に限定されるものではなく、上記メッセージが発行された際に第3のオブジェクトを生成してもよく、あるいは前もって生成しておいてもよい。

【0039】

また、本発明のオブジェクト指向プログラム記憶媒体のうちの第4のオブジェクト指向プログラム記憶媒体は、

データと処理とを有する複数のオブジェクト、が格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リストIDと、自オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リストIDとの対応をあらわす入力データタグテーブルを生成する入力データタグテーブル生成手段と、を

含むオブジェクト結合プログラムとが格納されてなることを特徴とする。

【0040】

ここで、上記本発明の第4のオブジェクト指向プログラム記憶媒体において、
上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

上記入力データタグテーブル生成手段が、入力データタグテーブルを生成するとともに、その入力データタグテーブルを、その入力データタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることが好ましい。

【0041】

ここで、入力データタグテーブルをメソッド要素に追加する態様の1つとして、そのメソッド要素には、直接的には、その入力データタグテーブルへのポインタを書き込んでもよい。

入力データタグテーブルが追加されてなるメソッド要素を持つ第1のオブジェクトは、そのメソッド要素により特定される第2のオブジェクトを呼び出すにあたり、そのメソッド要素に格納されたメソッドIDと入力データタグテーブルとを引数として第2のオブジェクトを呼び出すものであってもよい。

【0042】

ここで、入力データタグテーブルを引数として第2のオブジェクトを呼び出す態様の1つとして、直接的には、その入力データタグテーブルへのポインタを引数として第2のオブジェクトを呼び出してもよい。

このようにして第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行されると、第2のオブジェクトでは、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数であるメソッドIDにより特定されるメソッドが実行される。

【0043】

ここで、第2のオブジェクトは、上記のようにして第1のオブジェクトから第

2 のオブジェクトに向けてメッセージが発行された際、第 1 のオブジェクトから第 2 のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数である入力データタグテーブルを参照し第 1 のオブジェクトのポインタ要素リスト ID を取得して、第 1 のオブジェクトの、そのポインタ要素リスト ID により特定されるポインタ要素リストを生成するとともに、第 2 のオブジェクトの、そのポインタ要素リスト ID に対応するデータ要素リスト ID により特定されるデータ要素リストを生成し、第 1 のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、第 2 のオブジェクトのデータ要素リストに配列されたポインタを書き込むものであってもよい。

【0044】

あるいは、第 2 のオブジェクトが、第 3 のオブジェクトを生成する手段と、を有し、上記のようにして第 1 のオブジェクトから第 2 のオブジェクトに向けてメッセージが発行された際に、第 1 のオブジェクトから第 2 のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数である入力データタグテーブルを参照し第 1 のオブジェクトのポインタ要素リスト ID を取得して、第 1 のオブジェクトの、そのポインタ要素リスト ID により特定されるポインタ要素リストを生成するとともに、第 3 のオブジェクトの、そのポインタ要素リスト ID に対応するデータ要素リスト ID により特定されるデータ要素リストを生成し、第 1 のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、第 3 のオブジェクトのデータ要素リストに配列されたポインタを書き込むものであることも好ましい形態である。

【0045】

この第 4 のオブジェクト指向プログラム記憶媒体においても、第 2 のオブジェクトが第 3 のオブジェクトを生成するタイミングは特に限定されるものではなく、上記メッセージが発行された際に第 3 のオブジェクトを生成してもよく、あるいは前もって生成しておいてもよい。

さらに、本発明のオブジェクト指向プログラム記憶媒体のうちの第 5 のオブジェクト指向プログラム記憶媒体は、

データと処理とを有する複数のオブジェクトが格納されてなるとともに、

他オブジェクトに向けたメッセージの発行処理を行なう出力インストラクションバス部を持つ第1のオブジェクトと、他オブジェクトで発行された自オブジェクト向けのメッセージを受け付けて、自オブジェクトの、そのメッセージに対応するメソッドを起動する入力インストラクションバス部を持つ第2のオブジェクトとの間で、第1のオブジェクトのメッセージと第2のオブジェクトのメソッドとを対応づけることにより、第1のオブジェクトと第2のオブジェクトとの間でのメッセージの授受を可能ならしめるインストラクション結合手段と、

自オブジェクトの出力インストラクションバス部に、他オブジェクトそれぞれに対応する、他オブジェクトの、データへのポインタが格納されるポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リストIDと、自オブジェクトの、データが格納されるデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リストIDとの対応をあらわす出力データタグテーブルを生成する出力データタグテーブル生成手段と、を含むオブジェクト結合プログラムとが格納されてなることを特徴とする。

【0046】

ここで、上記本発明の第5のオブジェクト指向プログラム記憶媒体において、上記インストラクション結合手段が、自オブジェクトのメッセージに対応する他オブジェクトのメソッドを特定するメソッドIDと、そのメソッドIDにより特定されるメソッドが実行される他オブジェクトへのポインタとを含むメソッド要素の配列であるメソッド要素リストを生成するものであり、

上記出力データタグテーブル生成手段が、出力データタグテーブルを生成するとともに、その出力データタグテーブルを、その出力データタグテーブルに対応する他オブジェクトへのポインタを含むメソッド要素に追加するものであることが好ましい。

【0047】

ここで、出力データタグテーブルをメソッド要素に追加する態様の1つとして、そのメソッド要素には、直接的には、その出力データタグテーブルへのポインタを書き込んでもよい。

出力データタグテーブルが追加されてなるメソッド要素を持つ第1のオブジェ

クトは、そのメソッド要素により特定される第2のオブジェクトを呼び出すにあたり、そのメソッド要素に格納されたメソッドIDと出力データタグテーブルとを引数として第2のオブジェクトを呼び出すものであってもよい。

【0048】

ここで、出力データタグテーブルを引数として第2のオブジェクトを呼び出す態様の1つとして、直接的には、その出力データタグテーブルへのポインタを引数として第2のオブジェクトを呼び出してもよい。

このようにして第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行されると、第2のオブジェクトでは、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数であるメソッドIDに対応するメソッドが実行される。

【0049】

ここで、第2のオブジェクトは、上記のようにして第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行された際、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、そのメッセージの引数である出力データタグテーブルを参照し第1のオブジェクトのデータ要素リストIDを取得して、第1のオブジェクトの、そのデータ要素リストIDにより特定されるデータ要素リストを生成するとともに、第2のオブジェクトの、そのデータ要素リストIDに対応するポインタ要素リストIDにより特定されるポインタ要素リストを生成し、第2のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、第1のオブジェクトのデータ要素リストに配列されたポインタを書き込むものであってもよい。

【0050】

あるいは、第2のオブジェクトが、第3のオブジェクトを生成する手段を有し、上記のようにして第1のオブジェクトから第2のオブジェクトに向けてメッセージが発行された際に、第1のオブジェクトから第2のオブジェクトに向けたメッセージを受け付け、メッセージの引数である出力データタグテーブルを参照し第1のオブジェクトのデータ要素リストIDを取得して、第1のオブジェクトの、そのデータ要素リストIDにより特定されるデータ要素リストを生成すると

もに、第3のオブジェクトの、そのデータ要素リストIDに対応するポインタ要素リストIDにより特定されるポインタ要素を生成し、第3のオブジェクトのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、第1のオブジェクトのデータ要素リストに配列されたポインタを書き込むものであることも好ましい形態である。

【0051】

この第5のオブジェクト指向プログラム記憶媒体においても、第2のオブジェクトが第3のオブジェクトを生成するタイミングは特に限定されるものではなく、上記メッセージが発行された際に第3のオブジェクトを生成してもよく、あるいは前もって生成しておいてもよい。

また、上記目的を達成する本発明のオブジェクト間ネットワーク表示方法は、電子化された画像情報に基づく画像を表示画面上に表示する画像表示装置の表示画面上に、オブジェクト指向プログラミングにより生成される複数のオブジェクトと、これら複数のオブジェクト間のデータおよび制御の流れを示す配線とを表示するオブジェクト間ネットワーク表示方法において、

上記表示画面が複数の升目に区切られてなる升目1つからなる表示領域内、もしくは隣接する複数の升目が連結されてなる表示領域それぞれに、1つのオブジェクトを表示するオブジェクト表示部分領域と、複数のオブジェクト間を結ぶ配線を表示する配線表示部分領域とを、隣接する2つの表示領域それぞれのオブジェクト表示部分領域どうしの間に配線表示部分領域が形成されるように定め、

上記表示画面上に、複数のオブジェクトそれぞれが、各表示領域内のオブジェクト表示部分領域に1つずつ配置されるとともに、これら複数のオブジェクトどうしを結ぶ配線が、複数の表示領域に跨って連なる配線表示部分領域に表示されてなる第1の画像を表示することを特徴とする。

【0052】

本発明のオブジェクト間ネットワーク表示方法によれば、整然と並んだオブジェクトの配置が得られ、しかもオブジェクトを表示する領域と配線を表示する領域を分けたためオブジェクトと配線が重なることのない見やすい表示を得ることができる。

ここで、上記本発明のオブジェクト間ネットワーク表示方法において、上記第 1 の画像を構成する複数のオブジェクトのうちの所定のオブジェクトが、その所定のオブジェクトよりも下層側の複数のオブジェクトとそれら複数のオブジェクト間を結ぶ配線とから成るサブネットワークで構成されるものであって、

上記第 1 の画像に代えて、その第 1 の画像内の上記所定のオブジェクトの表示に代わりその所定のオブジェクトのサブネットワークが表示されてなる第 2 の画像を表示するにあたり、上記第 1 の画像内の、上記所定のオブジェクトの表示領域よりも拡大された表示領域にサブネットワークが表示されるとともに、そのサブネットワークの表示領域の上下および左右に並ぶ表示領域がそれぞれ左右および上下に拡大された表示領域に変更され、さらに、サブネットワークの表示領域に対し斜めに位置する表示領域については第 1 の画像と同一の寸法で表示されてなる第 2 の画像を表示するものであることが好ましい。

【0053】

このような表示方法を採用することによって、サブネットワークと、そのサブネットワークの周囲のネットワークとの接続状態を容易に確認することが可能となる。

あるいは、上記本発明のオブジェクト間ネットワーク表示方法において、上記第 1 の画像を構成する複数のオブジェクトのうちの所定のオブジェクトが、その所定のオブジェクトよりも下層側の複数のオブジェクトとこれら複数のオブジェクト間を結ぶ配線とから成るサブネットワークで構成されるものであって、

上記第 1 の画像に代えて、その第 1 の画像内の上記所定のオブジェクトの表示に代わりその所定のオブジェクトのサブネットワークが表示されてなる第 2 の画像を表示するにあたり、上記第 1 の画像内の、上記所定のオブジェクトの表示領域よりも拡大された表示領域にサブネットワークが表示されるとともに、その第 2 の画像の周縁に位置する表示領域、およびその表示領域の、第 2 の画像の外周に接する辺の位置および寸法が、上記第 1 の画像の周縁に位置する表示領域、およびその表示領域の、第 1 の画像の外周に接する辺の位置および寸法と比べ、同一のオブジェクトが表示された表示領域であり、かつ外周に接する辺の位置および寸法も同一となるように、サブネットワークの表示領域を除く表示領域が第 1

の画像上の対応する表示領域と比べ変形してなる第 2 の画像を表示するものであってもよい。

【0 0 5 4】

この表示方法を採用することによって、サブネットワークとその周囲のオブジェクトのネットワークとの接続状態を容易に確認することができる。この表示方法の場合さらに、変形してはいるものの、サブネットワークを表示した状態で、サブネットワーク表示前（第 1 の画像）に表示されていたネットワーク全体を確認することができる。

【0 0 5 5】

また、本発明のオブジェクト間ネットワーク表示方法において、上記第 1 の画像を表示するにあたり、表示領域内のオブジェクト表示部分領域の形状および寸法が、その表示領域の形状および寸法に応じた形状および寸法に規格化されてなることが好ましい。

このような規格化により、一層見やすい表示画面となる。

【0 0 5 6】

さらに、本発明のオブジェクト間ネットワーク表示方法において、上記第 1 の画像を表示するにあたり、先ず複数のオブジェクトを表示し、次いで、これら複数のオブジェクト間を、データもしくは制御の流れの方向が所定長部分毎に繰り返し表示されてなる配線で連結している状態を表示することが好ましい。

このような配線を採用すると、オブジェクトが表示画面の外にはみ出している場合であっても、その配線のどちら側に入力または出力が存在するかを容易に判断することが可能となる。尚、配線終了後はこのような配線に代えて、通常の配線、例えば線の終端または両端のみに矢印をつけた配線に変更してもよい。

【0 0 5 7】

さらに、本発明のオブジェクト間ネットワーク表示方法において、上記第 1 の画像を表示するにあたり、中心線と、その中心線の両脇に沿う、中心線とは異なる表示態様の縁線とからなる配線であって、相互に交差する配線のうち、同一のデータもしくは制御の流れを表わす配線については中心線どうしが連続し、異なるデータもしくは制御の流れを表わす配線については一方の配線の中心線を他方

の配線の縁線に接する位置で分断してなる配線を表示することも好ましい形態である。

【0058】

このような配線を採用することにより、交差する配線が相互に接続されているのかあるいは単に交差しているだけなのかを容易に判断することができる。

このような配線を採用すると、オブジェクトが表示画面の外にはみ出している場合であっても、その配線のどちら側に入力または出力が存在するかを容易に判断することが可能となる。尚、配線終了後はこのような配線に代えて、通常の配線、例えば線の終端または両端のみに矢印をつけた配線に変更してもよい。

【0059】

さらに、本発明のオブジェクト間ネットワーク表示方法において、上記第1の画像を表示するにあたり、中心線と、その中心線の両脇に沿う、中心線とは異なる表示態様の縁線とからなる配線であって、相互に交差する配線のうち、同一のデータもしくは制御の流れを表わす配線については中心線どうしが連続し、異なるデータもしくは制御の流れを表わす配線については一方の配線の中心線を他方の配線の縁線に接する位置で分断してなる配線を表示することも好ましい形態である。

【0060】

このような配線を採用することにより、交差する配線が相互に接続されているのかあるいは単に交差しているだけなのかを容易に判断することができる。

また、上記目的を達成する本発明のオブジェクト指向プログラミング支援装置のうちの第1のオブジェクト指向プログラミング支援装置は、データと処理とを有するオブジェクト複数を、指示に応じて相互に結合するオブジェクト指向プログラミング支援装置において、

オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジ

ェクトであって1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する表示手段と、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、

オブジェクトの階層構造を構築する階層構造構築手段と、

オブジェクト結合手段に対しオブジェクト間を結ぶ配線を指示入力するとともに、階層構造構築手段に対し、オブジェクトの、階層構造上の配置位置を指示入力する操作子とを備え、

上記階層構造構築手段が、操作子からの指示に応じて、指定されたオブジェクトである本体オブジェクトを複写して分身オブジェクトを作成し、その分身オブジェクトを、その本体オブジェクトが配置された階層とは異なる階層に配置する手段を有し、

上記オブジェクト結合手段が、分身オブジェクトが配置された階層における配線において、操作子からの、分身オブジェクトと他オブジェクトとの間の配線の指示を受け付けるとともに、その分身オブジェクトとその分身オブジェクトに対応する本体オブジェクトとを1つのオブジェクトとした結合構造を構築するものであることを特徴とする。

【0061】

このような分身オブジェクトを作成し、結合構造を構築するにあたって分身オブジェクトとその本体オブジェクトとを1つのオブジェクトとした結合構造を構築することにより、1つのオブジェクトを所望の複数の階層に自由に配置して配線（結合の指示）を行なうことができ、異なる階層どうしのオブジェクト間の配線が容易になり、視覚的にも見やすい表示が可能となる。

【0062】

また、上記目的を達成する本発明のオブジェクト指向プログラミング支援装置のうちの第2のオブジェクト指向プログラミング支援装置は、データと処理とを有するオブジェクト複数を、指示に応じて相互に結合するオブジェクト指向プロ

グラミング支援装置において、

オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジェクトであって1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する表示手段と、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と

、

オブジェクトの階層構造を構築する階層構造構築手段と、

オブジェクト結合手段に対しオブジェクト間を結ぶ配線を指示入力するとともに、前記階層構造構築手段に対し、オブジェクトの、階層構造上の配置位置を指示入力する操作子とを備え、

上記オブジェクト結合手段が、操作子からの、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトの、他オブジェクトとの結合構造を解除して、置き換え前のオブジェクトの他オブジェクトとの結合構造を、置き換え後のオブジェクトに継承させるものであり、

上記階層構造構築手段が、操作子からの、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトが配置されていた階層に、置き換え前のオブジェクトに代えて、置き換え後のオブジェクトを配置するものであることを特徴とする。

【0 0 6 3】

オブジェクトを置き換えるには、通常であれば、先ず置き換え前のオブジェクトの配線を外し、置き換え後のオブジェクトに対しあらためて配線を行なうことになるが、置き換え前のオブジェクトの配線（結合関係）を置き換え後のオブジェクトにそのまま踏襲させることにより、置き換え後のオブジェクトと他オブジ

エクトの間の配線の手間が省け、オブジェクトの置き換えが極めて容易となり、その分オブジェクト指向プログラミングが容易化される。

【0064】

さらに、上記目的を達成する本発明の第3のオブジェクト指向プログラミング支援装置は、データと処理とを有するオブジェクト複数を、指示に応じて相互に結合するオブジェクト指向プログラミング支援装置において、

オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジェクトであって1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する表示手段と、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、

オブジェクトの階層構造を構築する階層構造構築手段と、

オブジェクト結合手段に対しオブジェクト間を結ぶ配線を指示入力するとともに、階層構造構築手段に対し、オブジェクトの、階層構造上の配置位置を指示入力する操作子とを備え、

上記階層構造構築手段が、操作子からの、所定の階層に配置されたオブジェクトの中から複数のオブジェクトを指定してそれら複数のオブジェクトを一段下の階層に再配置する旨の指示に応じて、それら複数のオブジェクト相互間の結合構造、およびそれら複数のオブジェクトと、それら複数のオブジェクト以外の他オブジェクトとの間の結合構造が保持されるように、それら複数のオブジェクトをそれら一段下の階層に再配置するとともに上記所定の階層にそれら複数のオブジェクトを内包するオブジェクトを生成して配置するものであることを特徴とする。

【0065】

上記のように、複数のオブジェクトを、配線（結合関係）を保ったまま別の階層に配置替えすることができるようにするとプログラムを作成しながら作成したプログラムを整理することができ、また階層化された部分が1つのオブジェクトとして作用するため、そのオブジェクトを1つのプログラム部品と再利用することができる。

【0066】

さらに、本発明の第4のオブジェクト指向プログラミング支援装置は、データと処理とを有するオブジェクト複数を、指示に応じて相互に結合するオブジェクト指向プログラミング支援装置において、

オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジェクトであって1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する表示手段と、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、

オブジェクトの階層構造を構築する階層構造構築手段と、

オブジェクト結合手段に対しオブジェクト間を結ぶ配線を指示入力するとともに、階層構造構築手段に対し、オブジェクトの、階層構造上の配置位置を指示入力する操作子とを備え、

上記表示手段が、操作子による指示に応じて、指定された1つのメッセージ端子に接続されたメソッド端子が複数存在する場合の、これら複数のメソッド端子に対応する複数のメソッドの実行順序をあらわすリストを表示する手段を有し、

上記オブジェクト結合手段が、操作子による指示に応じて、表示手段に表示さ

れたリストにあらわれた複数のメソッドの実行順序が変更された結合構造を再構築する手段を有するものであることを特徴とする。

【0 0 6 7】

本発明の第 4 のオブジェクト指向プログラミング支援装置によれば、1 つのメッセージに対する複数のメソッドの実行順序を容易にかつ正確に知ることができ、またその実行順序を容易に変更することができる。

尚、本発明のオブジェクト指向プログラミング支援装置には、第 5 のオブジェクト指向プログラミング支援装置も存在するが、この第 5 のオブジェクト指向プログラミング支援装置については後述する。

【0 0 6 8】

また、上記目的を達成する本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第 1 のオブジェクト指向プログラミング用プログラム記憶媒体は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体において、

上記オブジェクトが、オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されるとともに、1 つのオブジェクトの内部に 1 つもしくは複数のオブジェクトの存在を許容する階層構造で表現されるオブジェクトであって、複数のオブジェクトの端子間を配線で結ぶ指示がなされるものであり、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、オブジェクトの階層構造を構築する階層構造構築手段とを含むプログラムであって、

上記階層構造構築手段が、指定されたオブジェクトである本体オブジェクトを複写して分身オブジェクトを作成し、その分身オブジェクトを、その本体オブジ

ェクトが配置された階層とは異なる階層に配置する手段を有し、上記オブジェクト結合手段が、分身オブジェクトが配置された階層における配線において、その分身オブジェクトと他のオブジェクトとの間の配線の指示を受け付けるとともに、その分身オブジェクトとその分身オブジェクトに対応する本体オブジェクトとを1つのオブジェクトとした結合構造を構築するものであるプログラムが格納されてなることを特徴とする。

【0069】

また、上記目的を達成する本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第2のオブジェクト指向プログラミング用プログラム記憶媒体は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体において、

上記オブジェクトが、オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されるとともに、1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されるオブジェクトであって、複数のオブジェクトの端子間を配線で結ぶ指示がなされるものであり、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、オブジェクトの階層構造を構築する階層構造構築手段とを含むプログラムであって、

上記オブジェクト結合手段が、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトの、他オブジェクトとの結合構造を解除して、置き換え前のオブジェクトの他オブジェクトとの結合構造を、置き換え後のオブジェクトに継承させるものであり、上記階層構造構築手段が、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトが配置されていた階層に、置き換え前の

オブジェクトに代えて、置き換え後のオブジェクトを配置するものであるプログラムが格納されてなることを特徴とする。

【0070】

さらに、本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第3のオブジェクト指向プログラミング用のプログラム記憶媒体は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体において、

上記オブジェクトが、オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されるとともに、1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されるオブジェクトであって、複数のオブジェクトの端子間を配線で結ぶ指示がなされるものであり、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築するオブジェクト結合手段と、オブジェクトの階層構造を構築する階層構造構築手段とを含むプログラムであって、

上記階層構造構築手段が、所定の階層に配置されたオブジェクトの中から指定された複数のオブジェクトを一段下の階層に再配置する旨の指示に応じて、それら複数のオブジェクト相互間の結合構造、およびそれら複数のオブジェクトと、それら複数のオブジェクト以外他オブジェクトとの間の結合構造が保持されるように、その複数のオブジェクトを一段下の階層に再配置するとともにそれら所定の階層にそれら複数のオブジェクトを内包するオブジェクトを生成して配置するものであるプログラムが格納されてなることを特徴とする。

【0071】

さらに、本発明のオブジェクト指向プログラミング用プログラム記憶媒体のう

ちの第4のオブジェクト指向プログラミング用のプログラム記憶媒体は、データと処理とを有するオブジェクト複数を相互に結合するオブジェクト指向プログラミングを支援するプログラムが格納されたオブジェクト指向プログラミング用プログラム記憶媒体において、

上記オブジェクトが、オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されるとともに、1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されるオブジェクトであって、複数のオブジェクトの端子間を配線で結ぶ指示がなされるものであり、

複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、該配線に対応した結合構造を構築するオブジェクト結合手段と、オブジェクトの階層構造を構築する階層構造構築手段とを含むプログラムであって、

上記オブジェクト結合手段が、指定された1つのメッセージ端子に接続されたメソッド端子が複数存在する場合の、これら複数のメソッド端子に対応する複数のメソッドの実行順序をあらわすリストを作成する手段と、そのリストにあらわれた複数のメソッドの実行順序の変更指示に応じてそれら複数のメソッドの実行順序が変更された結合構造を再構築する手段とを有するものであるプログラムが格納されてなることを特徴とする。

【0072】

また、本発明のコンポーネント記憶媒体のうちの第1のコンポーネント記憶媒体は、他オブジェクトで発行されたメッセージにより発火して所定の既存ソフトウェアのイベントを発行するメソッドを含む、該既存ソフトウェアと合わせて1つのオブジェクトとして作用するコンポーネントが格納されてなることを特徴とする。

【0073】

このようなコンポーネントにより、既存ソフトウェアがいわば‘包み込まれる

形となり、既存ソフトウェアの仕組みによらず、また、この既存ソフトウェアを改変することなく、既存ソフトウェアをオブジェクトとして取り込むことができ、既存ソフトウェアの再利用性が格段に向上する。

ここで、上記コンポーネントが、上記メソッドを含むとともにそのメソッドの実行により上記イベントが発行されたことを他オブジェクトに向けて通知するメッセージを含むものであることが好ましい。

【0074】

この場合、メソッドとメッセージとの間の接続による連鎖的な処理が可能となる。

また、本発明のコンポーネント記憶媒体のうちの第2のコンポーネント記憶媒体は、所定の既存ソフトウェアの所定のイベントの発生を受けてそのイベントが発生したことを他オブジェクトに向けて通知するメッセージを含む、その既存ソフトウェアと合わせて1つのオブジェクトとして作用するコンポーネントが格納されてなることを特徴とする。

【0075】

このようなコンポーネントにより、既存ソフトウェアを‘包み込む’ことにより、既存ソフトウェアに対するイベントが発生したのを受けて他オブジェクトのメソッドを連動して実行させるなどといった拡張機能を、既存ソフトウェア自体の拡張性に依存せずに実現することができる。

また、本発明のコンポーネント作成装置は、

メソッドの作成およびメッセージの作成を選択的に指示入力する第1の操作子と、

所定の既存ソフトウェアの所望のイベントの発行を指示入力する第2の操作子と、

既存ソフトウェアと合わせて1つのオブジェクトとして作用するコンポーネントを作成するコンポーネント作成手段であって、第1の操作子の操作によりメソッドの作成が指示され次いで第2の操作子の操作により既存ソフトウェアの所定のイベントが発行された場合に、そのコンポーネントに、他オブジェクトで発行されたメッセージにより発火してそのイベントを発行するメソッドを作成し、第

1の操作子によりメッセージの作成が指示され次いで第2の操作子により既存ソフトウェアの所定のイベントの発行が指示された場合に、そのコンポーネントに、そのイベントの発生に応答してそのイベントが発生したことを他オブジェクトに通知するためのメッセージを作成するコンポーネント作成手段とを備えたことを特徴とする。

【0076】

このようなコンポーネント作成装置を用いることによって、上述の第1のコンポーネント記憶媒体や第2のコンポーネント記憶媒体に格納されるコンポーネントを、プログラミングについての深い知識を要求することなく、対話的に容易に作成することができる。

また、本発明のオブジェクト指向プログラミング支援装置のうちの第5のオブジェクト指向プログラミング支援装置は、

他オブジェクトで発行されたメッセージにより発火して所定の既存ソフトウェアのイベントを発行するメソッドと、そのメソッドの実行により上記イベントが発行されたことを他オブジェクトに向けて通知するメッセージとを含みその既存ソフトウェアと合わせて1つのオブジェクトとして作用するコンポーネントを、1つもしくは複数の既存ソフトウェアに関して格納するコンポーネントファイルと、

既存ソフトウェアについてイベントの発行を指示入力するための操作子と、

その操作子の操作に応じて順次発行された、1つもしくは複数の既存ソフトウェアに関するイベントのリストを格納するイベントログファイルと、

イベントログファイルから順次にイベントを取り出して、前回取り出したイベントと同一のイベントが発行されたことを他オブジェクトに向けて通知するメッセージを含むコンポーネントのメッセージと、今回取り出したイベントと同一のイベントを発行するメソッドを含むコンポーネントのメソッドとを結合するコンポーネント結合手段とを備えたことを特徴とする。

【0077】

本発明の第5のオブジェクト指向プログラミング支援装置によれば、1つもしくは複数の既存ソフトウェアの複数のイベントの発行を実際に動作させたい順に

順次指示することにより、既存ソフトウェアをコンポーネントで包み込んだオブジェクト間のメッセージとメソッドとが結合され、プログラミングについての深い知識を要求することなく、既存アプリケーションの複数のイベントの自動操作が可能となる。

【0078】

【発明の実施の形態】

以下、本発明の実施形態について説明する。

ここでは、本発明の実施形態を総合したオブジェクトウェアプログラミングシステムの概要について先ず説明し、その後個々の実施形態の説明に移る。

図1は、本発明のオブジェクト指向プログラミング装置、オブジェクト指向プログラミング支援装置、コンポーネント作成装置、オブジェクト指向プログラム記憶媒体、オブジェクト指向プログラミング用プログラム記憶媒体、コンポーネント記憶媒体、およびオブジェクト間ネットワーク表示方法の各実施形態を包含するコンピュータシステムの外観図である。

【0079】

図1に示すコンピュータシステム100は、CPUやMO（光磁気ディスク）ドライブ装置等が内蔵された本体部101、本体部101からの指示により、表示画面102a上に画像を表示する画像表示部102、このコンピュータシステム100に種々の情報を入力するためのキーボード103、画像表示部102の表示画面102a上の任意の位置を指定するマウス104、以下において説明するオブジェクトやオブジェクト結合プログラム等を格納しておく記憶部105が備えられている。

【0080】

プログラムの開発は、この図1に示すコンピュータシステムでも行なうことができ、また、他の同種のコンピュータシステムを用いて開発されたものを可搬型記憶媒体であるMO（光磁気ディスク）110に格納し、そのMO110をこの図1に示すコンピュータシステム100に装填して、このコンピュータシステム100に入力することもできる。またそれと同様に、この図1に示すコンピュータシステム100を用いて開発したプログラムを、MO110を介して他のコン

ピュータシステムに移すこともできる。

【0081】

図2は、図1に示すコンピュータシステム内に実現されたオブジェクトウェアプログラミングシステムの概要図である。

このオブジェクトウェアプログラミングシステム120は、オブジェクトや、既存ソフトウェアを‘包み込む’コンポーネントを作成するオブジェクトビルダ部121、オブジェクト間の配線（結合関係）を表示し、編集するオブジェクト間配線エディタ部122、および、オブジェクトビルダ部121で作成されたオブジェクト（既存ソフトウェアとコンポーネントとの組合せからなるオブジェクトを含む）を、オブジェクト間配線エディタ部122で定義されたオブジェクト間の配線（接続関係）のとおり接続して実行するインタプリタ部123から構成されている。

【0082】

オブジェクトビルダ部121では、図1に示すコンピュータシステム100のキーボード103やマウス104の操作によりオブジェクトを最初から作成することもできるが、このオブジェクトウェアプログラミングシステム120には、このオブジェクトウェアプログラミングシステム120とは全く独立に、かつ、各種の異なるプログラム言語で開発された既存の各種のアプリケーションプログラム（以下、アプリケーションプログラムをアプリケーションと略記することがある）が格納された既存アプリケーションファイル131が備えられており、このオブジェクトビルダ121では、この既存アプリケーションファイル131に格納された既存アプリケーションを‘包み込み’、その既存アプリケーションと合わせて1つのオブジェクトとして作用するコンポーネントを作成することもできる。ここでは、特に断らない限り、このコンポーネントとそのコンポーネントにより‘包み込まれた’既存アプリケーションとを合わせたオブジェクトを含めて、オブジェクトと表現する。

【0083】

オブジェクトビルダ部121で作成されたオブジェクトは、オブジェクトデータファイル132と実行型オブジェクトファイル133に格納される。オブジェ

クトデータファイル 1 3 2 には、オブジェクトビルダ部 1 2 1 で作成されたオブジェクトをあらわす情報のうち、以下において説明するオブジェクト間配線エディタ部 1 2 2 における、オブジェクトの表示およびオブジェクト間の配線（結合関係の定義）に必要な情報が抽出されて格納される。一方、実行型オブジェクトファイル 1 3 3 には、オブジェクトビルダ部 1 2 1 で作成されたオブジェクトがコンパイラにより実行形式に変換され、その実行形式に変換されたオブジェクトが格納される。

【 0 0 8 4 】

オブジェクト間配線エディタ部 1 2 2 は、オブジェクトデータファイル 1 3 2 に格納されたオブジェクトの情報を受け取って図 1 に示す画像表示部 1 0 2 の表示画面 1 0 2 a にそのオブジェクトを模式的に表示し、キーボード 1 0 3 やマウス 1 0 4 の操作に応じてオブジェクト間の結合状態を定義する。ここでは、後述するように、表示画面 1 0 2 a 上に、ハードウェアとしての L S I （ L a r g e S c a l e I n t e g r a t e d C i r c u i t ）の表示形式に近い表示形式で表示し、それら複数の L S I の端子どうしを信号線で配線するような感覚でオブジェクト間の結合状態の定義が行なわれる。このため、以下では、オブジェクトのことを「L S I」と称し、オブジェクト間の結合状態を定義することを「配線」と称する場合がある。

【 0 0 8 5 】

オブジェクト間配線エディタ部 1 2 2 でオブジェクト間の配線を行なう際は、配線の途中結果をセーブしておき、またその途中結果をロードして表示することなどの目的で、オブジェクト間配線データファイル 1 3 4 が使用される。このオブジェクト間配線データファイル 1 3 4 には、マン・マシンインターフェイスとして都合の良い配線情報が格納される。例えばこのシステムでは、配線を行なうユーザの配線の理解の容易のためにオブジェクトが階層構造化されるが、オブジェクト間配線データファイル 1 3 4 には、そのような階層構造に関するデータも格納される。

【 0 0 8 6 】

このようにして、オブジェクト間配線エディタ部 1 2 2 で配線が完了すると、

オブジェクト間の結合状態をあらわす情報（以下、これを「配線データ」と称することがある）は、インタプリタ用配線データファイル 1 3 5 に格納される。配線データをインタプリタ用配線データファイル 1 3 5 に格納するにあたっては、例えばオブジェクトの階層構造の情報など、単にユーザの理解のための情報は捨象され、オブジェクト（ソフトウェア）を実際に動作させるにあたって必要となる配線データのみが抽出されてインタプリタ用配線データファイル 1 3 5 に格納される。

【 0 0 8 7 】

インタプリタ部 1 2 3 では、実行型オブジェクトファイル 1 3 3 に格納された実行形式のオブジェクトがインタプリタ用配線データファイル 1 3 5 に格納された配線データのとおり結合されて実行される。

以下、各実施形態について説明するが、説明の都合上および理解の容易のために、図 2 に示すオブジェクトウェアシステム 1 2 0 全体の中の位置付けからすると、先ずインタプリタ部 1 2 3 およびその周辺部に関連した実施形態について説明し、次いでオブジェクト間配線エディタ部 1 2 2 およびその周辺部に関連して実施形態について説明し、最後にオブジェクトビルダ部 1 2 1 およびその周辺部に関連した実施形態について説明する。

【 0 0 8 8 】

以下、先ず、インタプリタ部 1 2 3 およびその周辺部に関連した実施形態について説明する。

図 3 は、図 1 に示すコンピュータシステム内に実現されたソフトウェアシステムの第 1 例を示す模式図である。ここでは、この図 3 を参照して本発明の第 1 のオブジェクト指向プログラミング装置および本発明の第 1 のオブジェクト指向プログラム記憶媒体の一実施形態の全体構成について概括し、その後、図 4 以降の図を参照してその内容について詳細に説明する。

【 0 0 8 9 】

この図 3 に示すソフトウェアシステムと本発明との対応関係は以下のとおりである。すなわち、この図 3 に示すソフトウェアシステムが記憶された状態の記憶部 1 0 5（図 1 参照）が本発明の第 1 のオブジェクト指向プログラム記憶媒体の

一実施形態に相当し、図 1 に示すコンピュータシステム 100 のハードウェアと、そのコンピュータシステム 100 内で動作可能な状態にあるオブジェクト結合部 10 の組合せが、本発明の第 1 のオブジェクト指向プログラミング装置に相当する。尚、この図 3 に示すソフトウェアシステムが MO 110 にダウンロードされたときは、この図 3 に示すソフトウェアシステムを格納した状態にある MO 110 も、本発明の第 1 のオブジェクト指向プログラム記憶媒体の一例に相当する。

【0090】

ここでは、代表的に、それぞれがデータと処理（メソッド）とからなる 2 つのオブジェクト A、B を考える。

オブジェクト結合部 10 を構成する出力インストラクションバス部生成部 11 は、オブジェクト（ここでは代表的にオブジェクト A とする）の、他オブジェクト（ここでは代表的にオブジェクト B とする）に対するメッセージの発行処理を行なう出力インストラクションバス部の核となる部分を生成する。

【0091】

また、オブジェクト結合部 10 を構成する入力インストラクションバス部生成部 12 は、オブジェクト（ここでは代表的にオブジェクト B とする）の入力インストラクションバス部を生成する。入力インストラクションバス部は、他オブジェクト（ここでは代表的にオブジェクト A とする）で発行された自オブジェクト（ここではオブジェクト B）向けのメッセージを受け付けて、自オブジェクト（オブジェクト B）の、その受け付けたメッセージに対応するメソッドを起動する部分である。

【0092】

尚、本実施形態では、オブジェクト結合部 10 に出力インストラクションバス部生成部 11 および入力インストラクションバス部生成部 12 が備えられているが、オブジェクト A、B が、出力インストラクションバス部ないし入力インストラクションバス部に相当する構造をもともともっているものであってもよく、オブジェクト結合部 10 には、必ずしも出力インストラクションバス部生成部 11 および入力インストラクションバス部生成部 12 は備えられていなくてもよい。

【0093】

オブジェクト結合部10を構成するインストラクション結合部13は、オブジェクト相互間（ここでは、代表的にオブジェクトA，B間）で、オブジェクトAのメッセージとオブジェクトBのメソッドとを対応づける経路13aを生成することにより、オブジェクトAとオブジェクトBとの間でのメッセージの授受を可能ならしめる。

【0094】

オブジェクト結合部10を構成するデータ要素リスト生成部14は、オブジェクト（ここでは代表的にオブジェクトAとする）の、データが格納されるデータ格納領域へのポインタが配列されてなるデータ要素リストを生成する。

また、これと同様に、オブジェクト結合部10を構成するポインタ要素リスト生成部15は、オブジェクト（ここでは代表的にオブジェクトBとする）の、データへのポインタが格納されるポインタ格納領域へのポインタが配列されてなるポインタ要素リストを生成する。

【0095】

オブジェクト結合部10を構成するデータ結合部16は、オブジェクトA，B間で、ポインタ要素リスト生成部15により生成されたオブジェクトBのポインタ要素リストに配列されたポインタが指し示すポインタ格納領域に、データ要素リスト生成部14により生成されたオブジェクトAのデータ要素リストに配列されたポインタを書き込むことにより、オブジェクトA，B間でのデータの授受の経路16aを生成し、それらのオブジェクトA，B間でのデータの授受を可能ならしめる。

【0096】

図4は、図3に示すオブジェクトAの出力インストラクションバス部とオブジェクトBの入力インストラクションバス部のデータ構造の一例を示す模式図である。

オブジェクトAには、オブジェクトAのメッセージの最大数 $MA_A \text{ MAX}$ の配列からなるメッセージテーブルが備えられており、このメッセージテーブルには、各メッセージのメッセージ番号 MA_A （ここではメッセージ番号をMAであらわ

し、オブジェクトAのメッセージのメッセージ番号であることを添字Aであらわす)に対応して、以下に説明するメソッド要素リストへのポインタが格納されている。

【0097】

メソッド要素リストは、1つもしくは複数のメソッド要素の配列からなり、各メソッド要素は、メソッドを特定するメソッド番号MEと、そのメソッド番号MEにより特定されるメソッドが実行されるオブジェクトへのポインタと、次のメソッド要素へのポインタとにより構成されている。ここでは、メソッド番号をMEであらわし、そのメソッド番号MEにより特定されるメソッドが実行されるオブジェクトを添字であらわす。すなわち、図4に示す、最上段のメソッド要素には、オブジェクトBのメソッドのメソッド番号ME_Bと、オブジェクトBへのポインタが格納されている。

【0098】

最後段のメソッド要素の、次のメソッド要素へのポインタの欄には、自分が最後段であって自分の後ろに並ぶメソッド要素がないことをあらわす情報（ここではこの情報を‘ヌル’と称する）が格納されている。

このメソッド要素リストは、最も多くは、オブジェクトAのメッセージの数に対応した数だけ生成され、各メソッド要素リストは、オブジェクトAの各メッセージに対応しており、そのメッセージを発行する際に対応するメソッド要素リストが参照される。

【0099】

尚1つのメッセージに対し1つのメソッド要素リストが対応しているが、1つのメソッド要素リストに配列されたメソッド要素は、ある1つのオブジェクト（例えばオブジェクトB）に関するメソッド要素のみとは限らず、1つのメソッド要素リストには、複数のオブジェクトの複数のメソッドに関するメソッド要素が配列され得る。

【0100】

また、ここでは、オブジェクトAの出力インストラクションバス部の構造について説明しているが、出力インストラクションバス部は、他のオブジェクトに対

しメッセージを発行するオブジェクトそれぞれが持っている。

オブジェクトBには、オブジェクトBのメソッド ME_B の最大数 $ME_B\ MAX$ の配列からなるメソッドテーブルが備えられており、このメソッドテーブルには、各メソッドのメソッド番号 ME_B に対応して、そのメソッド番号 ME_B で特定されるメソッドへのポインタが格納されている。

【0101】

尚、ここでは、オブジェクトBの入力インストラクションバス部の構造について説明しているが、入力インストラクションバス部は、出力インストラクションバス部と同様に、他のオブジェクトで発行されたメッセージを受け付けて、そのメッセージに対応するメソッドを実行するオブジェクトそれぞれが持っている。また1つのオブジェクトが出力インストラクションバス部と入力インストラクションバス部との双方を持つことも多々あり得ることである。

【0102】

図5は、メッセージの発行処理を示すフローチャートである。

オブジェクトAで実行中のある処理においてメッセージを発行しようとしたとき、メッセージテーブルが参照され、その発行しようとしているメッセージのメッセージ番号 MA_A から、そのメッセージ番号 MA_A に対応するメソッド要素リストへのポインタが取得され（ステップ5__1）、そのポインタにより指示されたメソッド要素リストに配列された各メソッド要素がそれぞれ参照される。例えば図4の最上段のメソッド要素が参照されると、そのメソッド要素に格納されたポインタにより指示されるオブジェクトBが、そのメソッド要素に格納されたメソッド番号 ME_B を引数として呼び出される（ステップ5__2）。このようなメッセージ発行処理が、1つのメッセージの発行に対し、1つのメソッド要素リストに並ぶ各メソッド要素に関して行なわれる（ステップ5__3，5__4）。

【0103】

メソッド番号 ME_B を引数として呼び出されたオブジェクトBでは、その引数であるメソッド番号 ME_B を取得し（ステップ5__5）、メソッドテーブルを参照して、その取得したメソッド番号 ME_B により特定されるメソッドへのポインタを取得し、そのポインタにより指示されるメソッドの処理が行なわれる（ステ

ップ5__5)。

【0104】

図6は、図3に示すオブジェクト結合部10の出力インストラクションバス部生成部11の処理を示すフローチャートである。

ここでは、図4に示すMA_A MAX 幅のメッセージテーブルの枠が生成される（ステップ6__1）。

尚、本実施形態では、オブジェクトAがメッセージを発行する際、メッセージテーブルを介してメソッド要素リストのポインタを知るように構成されているが、例えば、オブジェクトAの処理（メソッド）内に直接にメソッド要素へのポインタが書き込まれていてもよく、その場合メッセージテーブルは不要である。すなわち、本発明においては、図6に示す処理、すなわち、図3に示す出力インストラクションバス部生成部11は必ずしも必要ではない。

【0105】

図7は、図3に示すオブジェクト結合部10の入力インストラクションバス部生成部12の処理を示すフローチャートである。

ここでは、図4に示す、ME_B MAX 幅のメソッドテーブルの枠が生成され（ステップ7__1）、その枠内の各メソッド番号ME_B の欄に、各メソッド番号ME_B に対応するメソッドへのポインタが格納される（ステップ7__2）。

【0106】

尚、本実施形態では、メソッドテーブルを介してメソッドへのポインタを認識するように構成しているが、メソッド番号ME_B とメソッドへのポインタとをメソッドテーブルの形式で対応づける必要はかならずしも必要ではなく、したがって図7に示す処理、すなわち図3に示す入力インストラクションバス部生成部12は必ずしも必要ではない。

【0107】

図8は、図3に示すオブジェクト結合部10におけるインストラクション結合部13の処理を示すフローチャートである。ここでも、オブジェクトBを他オブジェクトの代表としている。

ここでは、メソッド要素が生成されるが、このメソッド要素の生成にあたって

は、図 1 に示すコンピュータシステムを操作する操作者によりメッセージとメソッドとの対応関係が指定される。この対応関係は、

- (a) オブジェクト A のポインタ
- (b) オブジェクト B のポインタ
- (c) オブジェクト A のメッセージ番号 MA_A
- (d) オブジェクト B のメソッド番号 ME_B

の指定により定められる。尚、上記 (a) ~ (d) は直接的なアドレスや番号を操作者が指定するのではなく、そのオブジェクトの名前やそのオブジェクトでの処理（例えば表計算プログラム、表計算結果を画面に表示せよ）といった指定を、例えば表示画面 1 0 2 a（図 1 参照）に表示されたアイコンを、マウス 1 0 4 を操作してクリックすること等によりその指定が行なわれる。さらに詳細には、後述するように、オブジェクトを L S I の形式で表示しており、マウス 1 0 4 を用いて L S I の端子どうしを配線する操作を行なうことによりその指定が行なわれる。

【0 1 0 8】

ここに示す図 8 の処理では、先ずメソッド要素の枠が生成され（ステップ 8 __ 1）。そのメソッド要素の枠内に、メソッド番号 ME_B とオブジェクト B へのポインタが格納され（ステップ 8 __ 2）、対応するメッセージ番号 MA_A のメソッド要素リストに追加される（ステップ 8 __ 3）。すなわち、そのメソッド要素リストに並ぶ最終段のメソッド要素の、次のメソッド要素へのポインタの欄に、追加されるメソッド要素へのポインタが格納され、追加されるメソッド要素の、次のメソッド要素へのポインタの欄に‘ヌル’が格納される。必要に応じてこの図 8 に示す処理が繰り返され、データ要素リストが生成される。

【0 1 0 9】

尚、そのメソッド要素リストには未だ 1 つもメソッド要素が並んでいないときは、本実施形態では、メッセージテーブルの、対応するメッセージ番号 MA_A の欄に、登録しようとするメソッド要素へのポインタが格納される。

本実施形態では、このようにメソッド要素リストを作成することにより、オブジェクト A のメッセージとオブジェクト B のメソッドとを対応づけているため、

メッセージとメソッドとの対応関係が解りやすく、メッセージに対応したメソッドを直ちに認識することができ、高速処理が可能となる。

【0110】

図9は、図3に示すオブジェクトAのデータ要素リストのデータ構造の一例を示す模式図である。

オブジェクトAの内部には、オブジェクトBに受け渡す必要のある多数（ここではn個）のデータが存在し、図3に示すオブジェクト結合部10のデータ要素リスト生成部14により、図9に示すデータ要素リストが生成される。

【0111】

このデータ要素リストは、データの数nに対応した数のデータ要素の配列であり、各データ要素は、データが格納されるデータ格納領域へのポインタと、次のデータ要素へのポインタとにより構成されている。最終段のデータ要素の、次のデータ要素へのポインタの欄には‘ヌル’が書き込まれている。尚、図9において、例えばデータ格納領域1に対応するポインタをポインタ1と表記せずにポインタ1__1と表記したのは、後述するポインタとの区別のためである。

【0112】

図9における‘OUT_A’は、データ要素リスト番号を示している。データ要素リストは、データを受け渡す相手先の数などに応じて多数生成される可能性があり、ここでは、各データ要素リストはデータ要素リスト番号OUT_A（添字AはオブジェクトAをあらわす）で互いに識別されている。

図10は、図3に示すオブジェクト結合部10のデータ要素リスト生成部14の処理を示すフローチャートである。

【0113】

データ要素リストを生成するには、先ずデータ要素の枠が生成され（ステップ10__1）、その枠にデータ格納領域へのポインタが代入され（ステップ10__2）、データ要素リストに追加される（10__3）。データ要素リストに追加するにあたっては、そのデータ要素リストの最終段に既に並んでるデータ要素の、次のデータ要素へのポインタを格納する欄に、追加しようとするデータ要素リストへのポインタを格納され、追加しようとするデータ要素リストの、次のデータ

要素へのポインタが格納する欄に‘ヌル’が格納される。

【0114】

図10に示す処理が必要に応じて複数回繰り返されることにより、データ要素リストが生成される。

図11は、図3に示すオブジェクトBのポインタ要素リストのデータ構造の一例を示す模式図である。

オブジェクトBの内部には、オブジェクトAからデータを受け取る必要のある部分が多数（ここではn個）存在し、各部分には、ポインタ格納領域が確保されている。各ポインタ格納領域1～nには、データ要素とポインタ要素とが結合される前の現段階では、それぞれ勝手な、データへのポインタ1__3，2__3，…，n__3が格納されている。ここでは、図3に示すオブジェクト結合部10のポインタ要素リスト生成部15により、図11に示すポインタ要素リストが生成される。

【0115】

このポインタ要素リストは、ポインタ格納領域の数nに対応した数のポインタ要素の配列であり、各ポインタ要素は、対応するポインタ格納領域へのポインタと、次のポインタ要素へのポインタとにより構成されている。この図11では、例えばポインタ格納領域1へのポインタをポインタ1と表記せずにポインタ1__2と表記しており、さらにそのポインタ格納領域1に格納されている勝手なポインタをポインタ1__3と表記している。これは、図9に示すデータ要素に格納されたポインタを含めポインタを相互に区別するためである。

【0116】

ポインタ要素リストも、データ要素リストの場合と同様に、データを受け取る相手の数などに応じて多数生成される可能性があり、ここでは各ポインタ要素リストはポインタ要素リスト番号IN_B（添字BはオブジェクトBをあらわす）で互いに識別されている。

図12は、図3に示すオブジェクト結合部10のポインタ要素リスト生成部15の処理を示すフローチャートである。図10に示すデータ要素リスト生成部14の処理とはほぼ同様であり、簡単に説明するにとどめる。

【0117】

先ずポインタ要素の枠を生成し（ステップ12__1）、その枠内に、対応するポインタ格納領域へのポインタを格納し（ステップ12__2）、ポインタ要素リストに追加する（ステップ12__3）。この処理を必要に応じて複数回繰り返すことによりポインタ要素リストが生成される。

図13は、図3に示すオブジェクト結合部10のデータ結合部16の処理が実行された後の構造を示す模式図である。

【0118】

ここでは、オブジェクトBのポインタ格納領域1～nに、図9に示すデータ要素リストに配列されたデータ要素に格納されていたポインタ1__1～n__1が格納されており、オブジェクトAのデータをオブジェクトBから直接参照できる構造となっている。

図14は、図3に示すオブジェクト結合部10のデータ結合部16の処理を示すフローチャートである。

【0119】

先ず、図9に示すデータ要素リストの先頭に並ぶデータ要素に格納されたポインタ1__1が作業領域Pに格納され（ステップ14__1）、これと同様に、図11に示すポインタ要素リストの先頭に並ぶポインタ要素に格納されたポインタ1__2が作業領域Pに格納される（ステップ14__2）。

次に、Dが空か否か、すなわち、図9に示すデータ要素リストに並ぶ最終段のデータ要素まで、ステップ14__5において説明する対応付けが終了しているか否かが判定され（ステップ14__3）、Dが空のときは、この図14に示す処理を終了する。

【0120】

これと同様に、Pが空か否か、すなわち、図11に示すポインタ要素リストに並ぶ、最終段のポインタ要素まで対応付けが終了しているか否かが判定され（ステップ14__4）、Pが空のときは、この図14に示す処理を終了する。

ステップ14__5では、Pに格納されたポインタ（ここでは図11に示すポインタ1__2）が指し示す先のポインタ格納領域（ここではポインタ格納領域1）

に格納されているポインタ（ここではポインタ1__3）に代えて、Dに格納されたポインタ（ここでは図9に示すポインタ1__1）が代入される。これにより、図13に示す、オブジェクトAのデータ1とオブジェクトBのポインタ1__1との対応が成立する。

【0121】

ステップ14__6では、図9に示すデータ要素リストの、次に並ぶデータ要素に格納されたポインタ（ここではポインタ2__1）が作業領域Dに格納され、これとともに、図11に示すポインタ要素リストの、次に並ぶポインタ要素に格納されたポインタ（ここではポインタ2__2）が作業領域Pに格納され、ステップ14__3に戻る。これを繰り返し、ステップ14__5において、図9に示すデータ要素リストの最終段のデータ要素と、図11に示すポインタ要素リストの最終段のポインタ要素との対応付けが行なわれると、次のステップ14__6では作業領域Dと作業領域Pが空にされてステップ14__3に戻り、この図14に示す処理を終了する。ここではデータ要素リストに並ぶデータ要素の数とポインタ要素リストに並ぶポインタ要素の数が同数であるとして説明したが、それらの数に相違があったときは数の少ない方の対応づけが終了した時点でDもしくはPが空になり、この図14の処理を終了する。

【0122】

この図14の処理、すなわちデータ要素リストとポインタ要素リストとの対応づけが終了した後は、データ要素リストおよびポインタ要素リストは不要となり、消去される。

尚、図9～図14を参照して説明したデータ結合処理においては、図1に示すコンピュータシステム100の操作者により、

- (a) オブジェクトAのポインタ
- (b) オブジェクトBのポインタ
- (c) オブジェクトAのデータ要素リスト番号OUT_A
- (d) オブジェクトBのポインタ要素リスト番号IN_A

が入力される。尚、これらの情報の入力にあたっては、図8を参照して説明したメッセージとメソッドとの対応関係の入力の場合と同様に、操作者にとって解り

やすい表現であらわされた、表示画面102a（図1参照）上のアイコンのクリック等により、LSIを配線するイメージで行なわれる。

【0123】

図14の処理では、データ要素リストに並ぶデータ要素とポインタ要素リストに並ぶポインタ要素を、その配列順に対応づけを行なったが、例えばオブジェクトA、Bを作成する際に対応するデータ格納領域とポインタ格納領域に同一の名前、あるいは対応する名前を付けるというルール、あるいは、同一もしくは対応する名前の配列であって対応するものどうしが同一の順序となるように配列するというルールを定めておくことにより、その名前や配列の順序を参照して、互に対応するデータ要素とポインタ要素どうしが同一順に並んだデータ要素リストとポインタ要素リストを生成することができ、図14の処理に示すように配列順に対応づけることが可能となる。

【0124】

本実施形態によれば、図3に示すように、オブジェクトAのデータをオブジェクトBから直接参照することができ、したがってオブジェクト間のデータの受け渡しが極めて効率的に行なわれ、複数のオブジェクトに跨る処理動作速度が大幅に改善されることとなり、処理速度の低下を懸念して大きなオブジェクトを作成する必要がなく、小さな単位のオブジェクトを多数作成することができ、ソフトウェアの再利用性を大幅に向上させることができる。

【0125】

以上に説明した実施形態は、図3に示すオブジェクト結合部10により、初期化の段階、すなわち複数のオブジェクトからなるソフトウェアシステムを構築する段階で複数のオブジェクトを結合するものであり、その構築したソフトウェアシステムが動作を開始した後のオブジェクトどうしの再結合は考慮されていない。

【0126】

そこで、次に、上述した実施形態をベースにして、一旦構築したソフトウェアシステムが動作を開始した後、動的な再結合が行なわれる実施形態について説明する。

以下では、先ず図 15～図 18 を順次参照しながら、本発明の第 2～第 5 のオブジェクト指向プログラミング装置および本発明の第 2～第 5 のオブジェクト指向プログラム記憶媒体の各実施形態の全体構成について概括し、その後、図 19～図 35 を参照して、それら第 2～第 5 のオブジェクト指向プログラミング装置および第 2～第 5 のオブジェクト指向プログラム記憶媒体を総合した実施形態について詳細に説明する。

【0127】

図 15 は、図 1 に示すコンピュータシステム内に実現されたソフトウェアシステムの第 2 例を示す模式図である。

この図 15 に示すソフトウェアシステムと本発明との対応関係は以下のとおりである。すなわち、この図 15 に示すソフトウェアシステムが記憶された状態の記憶部 105（図 1 参照）が本発明の第 2 のオブジェクト指向プログラム記憶媒体の一実施形態に相当し、図 1 に示すコンピュータシステム 100 のハードウェアと、そのコンピュータシステム 100 内で動作可能な状態にあるオブジェクト結合部 20 との組合せが、本発明の第 2 のオブジェクト指向プログラミング装置の一実施形態に相当する。また、図 15 に示すソフトウェアシステムが図 1 に示す MO110 にダウンロードされたときは、図 15 に示すソフトウェアシステムを格納した状態にある MO110 も、本発明の第 2 のオブジェクト指向プログラム記憶媒体の一実施形態に相当する。

【0128】

この図 15 においても、多数のオブジェクトを代表して、2 つのオブジェクト A、B が示されている。

この図 15 に示すオブジェクト結合部 20 における、出力インストラクションバス部生成部 21、入力インストラクションバス部生成部 22、およびインストラクション結合部 23 の各処理は、図 3 に示すオブジェクト結合部 10 の、出力インストラクションバス部生成部 11、入力インストラクションバス部生成部 12、およびインストラクション結合部 13 の各処理と同一であり、インストラクション結合部 23 により、オブジェクト A のメッセージとオブジェクト B のメソッドとを対応づける経路 23a が生成される。出力インストラクションバス部生

成部21および入力インストラクションバス部生成部22は、本発明において必ずしも必要ではない点も前述と同様である。

入力インストラクションタグテーブル生成部24は、オブジェクトAの出力インストラクションバス部に、他オブジェクトそれぞれ（ここでは代表的にオブジェクトB）に対応する、オブジェクトBのメッセージとオブジェクトAのメソッドとの対応をあらわす入力インストラクションタグテーブルを生成する。

【0129】

この入力インストラクションタグテーブルは、後述するように、オブジェクトAからオブジェクトBに向けて発行されたメッセージの引数としてオブジェクトBに渡され、オブジェクトBでは、そのオブジェクトBの処理中において、いわば動的に、例えばオブジェクトBからオブジェクトAへのメッセージの発行の経路が生成される。

【0130】

図16は、図1に示すコンピュータシステムに実現されたソフトウェアシステムの第3例を示す模式図である。

この図16に示すソフトウェアシステムと本発明との対応関係は以下のとおりである。すなわち、この図16に示すソフトウェアシステムが記憶された状態の記憶部105（図1参照）が本発明の第3のオブジェクト指向プログラム記憶媒体の一実施形態に相当し、図1に示すコンピュータシステム100のハードウェアと、そのコンピュータシステム100内で動作可能な状態にあるオブジェクト結合部30との組合せが、本発明の第3のオブジェクト指向プログラミング装置の一実施形態に相当する。また、図16に示すソフトウェアシステムが図1に示すMO110にダウンロードされたときは、図16に示すソフトウェアシステムを格納した状態にあるMO110も、本発明の第3のオブジェクト指向プログラム記憶媒体の一実施形態に相当する。

【0131】

この図16においても、多数のオブジェクトを代表して、2つのオブジェクトA、Bが示されている。

この図16に示すオブジェクト結合部30における、出力インストラクション

バス部生成部 31、入力インストラクションバス部生成部 32、およびインストラクション結合部 33 の各処理は、図 3 に示すオブジェクト結合部 10 の、出力インストラクションバス部生成部 11、入力インストラクションバス部生成部 12、およびインストラクション結合部 13 の各処理と同一であり、インストラクション結合部 33 により、オブジェクト A のメッセージとオブジェクト B のメソッドとを対応付ける経路 33a が生成される。出力インストラクションバス部生成部 31 および入力インストラクションバス部生成部 32 は、本発明において必ずしも必要ではない点も前述と同様である。

【0132】

出力インストラクションタグテーブル生成部 34 は、オブジェクト A の出力インストラクションバス部に、他オブジェクトそれぞれ（ここでは代表的にオブジェクト B）に対応する、オブジェクト B のメソッドとオブジェクト A のメッセージとの対応をあらわす出力インストラクションタグテーブルを生成する。

この出力インストラクションタグテーブルは、後述するように、オブジェクト A からオブジェクト B に向けて発行されたメッセージの引数としてオブジェクト B に渡され、そのオブジェクト B の処理中において、いわば動的に、例えばオブジェクト A からオブジェクト B A へのメッセージの発行の経路が再編成される。

【0133】

図 17 は、図 1 に示すコンピュータシステムに実現されたソフトウェアシステムの第 4 例を示す模式図である。

この図 17 に示すソフトウェアシステムと本発明との対応関係は以下のとおりである。すなわち、この図 17 に示すソフトウェアシステムが記憶された状態の記憶部 105（図 1 参照）が本発明の第 4 のオブジェクト指向プログラム記憶媒体の一実施形態に相当し、図 1 に示すコンピュータシステム 100 のハードウェアと、そのコンピュータシステム 100 内で動作可能な状態にあるオブジェクト結合部 40 との組合せが、本発明の第 4 のオブジェクト指向プログラミング装置の一実施形態に相当する。また、図 17 に示すソフトウェアシステムが図 1 に示す MO 110 にダウンロードされたときは、図 17 に示すソフトウェアシステムを格納した状態にある MO 110 も、本発明の第 4 のオブジェクト指向プログラ

ム記憶媒体の一実施形態に相当する。

【0 1 3 4】

この図 1 7 においても、多数のオブジェクトを代表して、2つのオブジェクト A, B が示されている。

この図 1 7 に示すオブジェクト結合部 4 0 における、出力インストラクションバス部生成部 4 1、入力インストラクションバス部生成部 4 2、およびインストラクション結合部 4 3 の各処理は、図 3 に示すオブジェクト結合部 1 0 の、出力インストラクションバス部生成部 1 1、入力インストラクションバス部生成部 1 2、およびインストラクション結合部 1 3 の各処理と同一であり、インストラクション結合部 4 3 により、オブジェクト A のメッセージとオブジェクト B のメソッドとを対応付ける経路 4 3 a が生成される。出力インストラクションバス部生成部 4 1 および入力インストラクションバス部生成部 4 2 は、本発明において必ずしも必要ではない点も前述と同様である。

【0 1 3 5】

入力データタグテーブル生成部 4 4 は、オブジェクト A の出力インストラクションバス部に、他オブジェクトそれぞれ（ここでは代表的にオブジェクト B）に対応する、オブジェクト B のデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リスト番号 OUT_B と、オブジェクト A のポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リスト番号 IN_A との対応をあらわす入力データタグテーブルを生成する。

【0 1 3 6】

データ要素リスト番号 OUT_B およびポインタ要素リスト番号 IN_A は、それぞれ、オブジェクト B およびオブジェクト A を作成する段階で定められるが、この入力データタグテーブルが生成されただけの段階では、データ要素リスト自体およびポインタ要素リスト自体は未だ生成されていない。入力データタグテーブルはオブジェクト A からオブジェクト B に向けて発行されたメッセージの引数としてオブジェクト B に渡され、オブジェクト B では、そのオブジェクト B の処理中において、いわば動的に、自分（オブジェクト B）のデータ要素リストを生成するとともにオブジェクト A のポインタ要素リストを生成し、それらのデータ要

素リストとポインタ要素リストを結合する。詳細は後述する。

【0137】

図18は、図1に示すコンピュータシステムに実現されたソフトウェアシステムの第5例を示す模式図である。

この図18に示すソフトウェアシステムと本発明との対応関係は以下のとおりである。すなわち、この図18に示すソフトウェアシステムが記憶された状態の記憶部105（図1参照）が本発明の第5のオブジェクト指向プログラム記憶媒体の一実施形態に相当し、図1に示すコンピュータシステム100のハードウェアと、そのコンピュータシステム100内で動作可能な状態にあるオブジェクト結合部50との組合せが、本発明の第5のオブジェクト指向プログラミング装置の一実施形態に相当する。また、図18に示すソフトウェアシステムが図1に示すMO110にダウンロードされたときは、図18に示すソフトウェアシステムを格納した状態にあるMO110も、本発明の第5のオブジェクト指向プログラム記憶媒体の一実施形態に相当する。

【0138】

この図18においても、多数のオブジェクトを代表して、2つのオブジェクトA、Bが示されている。

この図18に示すオブジェクト結合部50における、出力インストラクションバス部生成部51、入力インストラクションバス部生成部52、およびインストラクション結合部53の各処理は、図3に示すオブジェクト結合部10の、出力インストラクションバス部生成部11、入力インストラクションバス部生成部12、およびインストラクション結合部13の各処理と同一であり、インストラクション結合部53により、オブジェクトAのメッセージとオブジェクトBのメソッドとを対応づける経路53aが生成される。インストラクションバス部生成部51および入力インストラクションバス部生成部52は、本発明において必ずしも必要ではない点も前述と同様である。

【0139】

出力データタグテーブル生成部54は、オブジェクトAの出力インストラクションバス部に、他オブジェクトそれぞれ（ここでは代表的にオブジェクトB）に

対応する、オブジェクト B のポインタ格納領域へのポインタが配列されるポインタ要素リストを特定するポインタ要素リスト番号 IN_B と、オブジェクト A のデータ格納領域へのポインタが配列されるデータ要素リストを特定するデータ要素リスト番号 OUT_A との対応をあらわす出力データタグテーブルを生成する。

【0140】

ポインタ要素リスト番号 IN_B およびデータ要素リスト番号 OUT_A は、それぞれ、オブジェクト B およびオブジェクト A を作成する段階で定められるが、この出力データタグテーブルが生成されただけの段階では、ポインタ要素リスト自体およびデータ要素リスト自体は未だ生成されていない。出力データタグテーブルはオブジェクト A からオブジェクト B に向けて発行されたメッセージの引数としてオブジェクト B に渡され、オブジェクト B では、そのオブジェクト B の処理中において、いわば動的に、オブジェクト A のデータ要素リストを生成するとともに、自分（オブジェクト B）のポインタ要素リストを生成し、それらのデータ要素リストとポインタ要素リストを結合する。詳細は後述する。

【0141】

図 19 は、図 15～図 18 に示すオブジェクト A の、データ構造の一部を示す模式図である。この図 19 には、図 4 に示すデータ構造のうちのメソッド要素 1 つ分と、そのメソッド要素に結合した入力インストラクションタグテーブル、出力インストラクションタグテーブル、入力データタグテーブル、および出力データタグテーブルが示されている。この図 19 には、各図 15～18 を参照してそれぞれ説明した各実施形態の特徴を全て備えた、総合的な実施形態のデータ構造が示されている。

【0142】

この図 19 に示すメソッド要素には、図 4 に示すメソッド要素の構成、すなわち、他オブジェクト（ここでは代表的にオブジェクト B）のメソッド番号 ME_B 、そのメソッド番号 ME_B により特定されるメソッドを実行するオブジェクト（ここではオブジェクト B）へのポインタ、および次のメソッド要素へのポインタに、さらに、入力インストラクションタグテーブルへのポインタ（以下、P1 と称することがある）、出力インストラクションタグテーブルへのポインタ（以下

、P2と称することがある）、入力データタグテーブルへのポインタ（以下、P3と称することがある）、および出力データタグテーブルへのポインタ（以下、P4と称することがある）、および自分自身（オブジェクトA）へのポインタ（以下、P5と称することがある）が付加されている。

【0143】

入力インストラクションタグテーブルは、他オブジェクト（ここではオブジェクトB）のメッセージの最大数 $MA_B \text{ MAX}$ と同じ幅の配列であって、オブジェクトBのメッセージ番号 MA_B に対応して、オブジェクトAのメソッド番号 ME_A が格納されている。

また、出力インストラクションタグテーブルは、他オブジェクト（ここではオブジェクトB）のメソッドの最大数 $ME_B \text{ MAX}$ と同じ幅の配列であって、オブジェクトBのメソッド番号 ME_B に対応して、オブジェクトAのメッセージ番号 MA_A が格納されている。

【0144】

入力データタグテーブル10は、他オブジェクト（ここではオブジェクトB）のデータ要素リストの最大数 $OUT_B \text{ MAX}$ と同じ幅の配列であって、オブジェクトBのデータ要素リスト番号 OUT_B に対応して、オブジェクトAのポインタ要素リスト番号 IN_A が格納されている。

さらに出力データタグテーブルは、他オブジェクト（ここではオブジェクトB）のポインタ要素リストの最大数 $IN_B \text{ MAX}$ と同じ幅の配列であって、オブジェクトBのポインタ要素リスト番号 IN_B に対応して、オブジェクトAのデータ要素リスト番号 OUT_A が格納されている。

【0145】

尚、この図19には、オブジェクトAの出力インストラクションバス部に関し、さらにオブジェクトBに関する4つのタグテーブルが示されているが、一般的には、これら4つのタグテーブルは、オブジェクトAの出力インストラクションバス部全体としては、オブジェクトAで発行するメッセージを受け取る相手のオブジェクトの数だけ保持される。すなわち、これら4つのタグテーブルは、相手のオブジェクトそれぞれに対応している。オブジェクトAに限らず他のオブジェ

クトの出力インストラクションバス部に関しても同様である。

【0146】

オブジェクトAが図19に示すメソッド要素を参照しオブジェクトBに向けてメッセージを発行するにあたっては、オブジェクトBのメソッド番号ME_Bのほか、必要に応じてP1～P5のうちの一部もしくは全部が引数としてオブジェクトBに渡される。あるいは、常に、メソッド番号ME_B、P1～P5の全てを引数としてもよい。

【0147】

以下、特に断らない限り、図4に示すデータ構造が図19に示すように変更されたデータ構造を前提にして説明する。

図20は、オブジェクトAのメッセージ発行処理の一例を示すフローチャートである。

ステップ20__1, 20__3, 20__4は、図5のステップ5__1, 5__3, 5__4と同一であり、説明は省略する。

【0148】

ステップ20__2では、メソッド番号ME_Bに加え、ここに示す例では、P1, P2, P5を引数としてオブジェクトBに渡され、このメッセージを受け取ったオブジェクトBでは、図5(B)に示すフローに従ってメソッド番号ME_Bにより特定されるメソッドの処理が実行される。

図21は、オブジェクトBの処理中における部分処理の第1例を示すフローチャートである。この部分処理は、引数としてオブジェクトBに渡されたメソッド番号ME_Bにより特定されるメソッドの処理中で実行される。

【0149】

ここでは、オブジェクトBに引数として渡された入力インストラクションバステーブルが参照され、オブジェクトBのメッセージ番号MA_BからオブジェクトAのメソッド番号ME_Aが取得され(ステップ21__1)、オブジェクトBの処理中で、その取得したオブジェクトAのメソッド番号ME_Aのメソッドの処理が実行される。

【0150】

図22は、オブジェクトBの処理中における部分処理の第2例を示すフローチャートである。

ここでは、引数としてオブジェクトBに渡された入力インストラクションテーブルが参照され、オブジェクトBのメッセージ番号MA_BからオブジェクトAのメソッド番号ME_Aが取得され（ステップ22__1）、自分（オブジェクトB）のメッセージテーブルのメッセージ番号MA_Bに対応するメソッド要素リストに、オブジェクトAのメソッド番号ME_Aに関するメソッド要素を追加する（ステップ22__2）。こうすることにより、以後、オブジェクトBでメッセージ番号MA_Bのメッセージを発行すると、オブジェクトAのメソッド番号ME_Aのメソッドが実行されるようになる。

【0151】

図23は、オブジェクトBの処理中における部分処理の第3例を示すフローチャートである。ただしこの部分処理は直接にはオブジェクトAで発行されたメッセージの引数は参照されていない。

ここでは、オブジェクトBの処理によりオブジェクトCが生成される（ステップ23__1）。1つのオブジェクト中で、他のオブジェクトを生成する処理自体は、オブジェクト指向プログラミングにおける通常の処理の1つであり、ここではオブジェクトCを生成する手法についての説明は省略する。

【0152】

図24は、オブジェクトBの処理中における部分処理の第4例を示すフローチャートである。

この図24に示す部分処理は、その実行に先立って図23に示す部分処理が実行され、オブジェクトCが生成されている必要がある。ただし、このオブジェクトCの生成タイミングは、特に限定されるものではなく、オブジェクトBにおける今回の一連処理中でオブジェクトCを生成してもよく、あるいはオブジェクトBにおける前回、あるいはそれ以前の処理においてオブジェクトCが生成されていてよい。

【0153】

図24の部分処理中では、引数としてオブジェクトBに渡された入力インスト

ラクションタグテーブルが参照され、もともとはオブジェクトBのメッセージ番号であって、オブジェクトCに引き継がれたメッセージ番号MA_Bに対応する、オブジェクトAのメソッド番号ME_Aが取得される（ステップ24__1）。そして、今度は、オブジェクトCのメッセージテーブルのメッセージ番号MA_Bに対応するオブジェクトCのメソッド要素リストに、オブジェクトAのメソッド番号ME_Aのメソッド要素が追加される（ステップ24__2）。これにより、オブジェクトCからオブジェクトAへのメッセージの経路が形成される。

【0154】

図25は、オブジェクトBの処理中における部分処理の第5例を示すフローチャートである。

ここでは出力インストラクションタグテーブルが参照されて、オブジェクトBのメソッド番号ME_Bに対応するオブジェクトAのメッセージ番号MA_Aが取得され（ステップ25__1）、オブジェクトAのメッセージテーブルのメッセージ番号MA_Aに対応するメソッド要素リストに、オブジェクトBのメソッド番号ME_Bのメソッド要素が追加される（ステップ25__2）。こうすることにより、以後、オブジェクトAでメッセージ番号MA_Aのメッセージが発行されると、オブジェクトBのメソッド番号ME_Bのメソッドが実行される。

【0155】

図26は、オブジェクトBの処理中における部分処理の第6例を示すフローチャートである。

この図26に示す部分処理を実行するには、それに先立って、オブジェクトCの生成処理（図23参照）が実行されている必要があるが、オブジェクトCの生成処理は、オブジェクトBにおける今回の一連の処理中で行われていてもよく、あるいは前回もしくはそれ以前の処理でオブジェクトCが生成されていてもよい。

【0156】

図26の部分処理では、引数としてオブジェクトBに渡された出力インストラクションタグテーブルが参照され、もともとはオブジェクトBのメソッド番号であってオブジェクトCに引き継がれたメソッド番号ME_Bに対応するオブジェク

トAのメッセージ番号 MA_A が取得され（ステップ26__1）、オブジェクトAのメッセージテーブルのメッセージ番号 MA_A に対応するメソッド要素リストに、メソッド番号 ME_B と、オブジェクトCへのポインタとが格納されたメソッド要素が追加される（ステップ26__2）。

【0157】

こうすることにより、これ以後、オブジェクトAから、新たに生成されたオブジェクトCに対しメッセージを発行することができるようになる。

図27は、オブジェクトAのメッセージ発行処理の、図20に示すメッセージ発行処理例とは異なる例を示すフローチャートである。

ステップ27__1, 27__3, 27__4は、図20のステップ20__1, 20__3, 20__4、および図5のステップ5__1, 5__3, 5__4と同様であり、説明は省略する。

【0158】

ステップ27__2では、メソッド番号 ME_B に加え、P3, P4, P5を引数として、オブジェクトBが呼び出される。

このメッセージを受け取ったオブジェクトBでは、メソッド番号 ME_B により特定されるメソッドの処理が実行される。

図28は、オブジェクトBの処理中における部分処理の第7例を示すフローチャートである。

【0159】

ここでは、引数としてオブジェクトBに引き渡された入力データタグテーブルが参照され、オブジェクトBのデータ要素リスト番号 OUT_B から、オブジェクトAのポインタ要素リスト番号 IN_A が取得され（ステップ28__1）、その取得したポインタ要素リスト番号 IN_A に対応する、オブジェクトAのポインタ要素リスト（図11参照。ただし、図11にはオブジェクトBのポインタ要素リストが示されている）が生成され（ステップ28__2）、データ要素リスト番号 OUT_B に対応する、オブジェクトBのデータ要素リスト（図9参照。ただし図9にはオブジェクトAのデータ要素リストが示されている）が生成され（ステップ28__3）、これらデータ要素リストとポインタ要素リストの結合処理（図13

参照。ただし、図13では、オブジェクトBのポインタがオブジェクトAのデータを指し示しており、ここでの説明と比べると、オブジェクトAとオブジェクトBの立場が逆である）が行なわれる（ステップ28__4）。

【0160】

このようにして、本実施形態では、処理の実行中に、いわば動的に、オブジェクト間でのデータ授受の経路が形成される。

図29は、オブジェクトBの処理中の部分処理の第8例を示すフローチャートである。

この図29に示す部分処理を実行するには、それに先立って、オブジェクトCの生成処理（図23参照）が実行されている必要がある。ただし、そのオブジェクトCの生成処理の実行は、この図29に示す部分処理以前であればいつでもよい。

【0161】

図29に示す部分処理では、引数としてオブジェクトBに引き渡された入力データタグテーブルが参照され、もともとはオブジェクトBのデータ要素リスト番号であってオブジェクトCに引き継がれたデータ要素リスト番号OUT_Bから、オブジェクトAのポインタ要素リスト番号IN_Aが取得され（ステップ29__1）、その取得したポインタ要素リスト番号IN_Aに対応する、オブジェクトAのポインタ要素リストが生成される（ステップ29__2）。さらに、データ要素リスト番号OUT_Bに対応する、オブジェクトCのデータ要素リストが生成され（ステップ29__3）、それらオブジェクトCのデータ要素リストと、オブジェクトAのポインタ要素リストとの結合処理が行なわれる（ステップ29__4）。

【0162】

このようにして、処理実行中に、いわば動的に、オブジェクトAから、新たに生成されたオブジェクトCのデータを直接に参照する経路が生成される。

図30は、オブジェクトBの処理中の部分処理の第9例を示すフローチャートである。

この図30に示す部分処理では、まず、オブジェクトBに引数として渡された出力データタグテーブルが参照され、オブジェクトBのポインタ要素リスト番号

IN_B から、オブジェクトAのデータ要素リスト番号OUT_A が取得される（ステップ30__1）。次いで、その取得したデータ要素リスト番号OUT_A に対応する、オブジェクトAのデータ要素リストが生成され（ステップ30__2）、また、ポインタ要素リスト番号IN_B に対応する、自分（オブジェクトB）のポインタ要素リストが生成され（ステップ30__3）、それらオブジェクトAのデータ要素リストとオブジェクトBのポインタ要素リストとの結合処理が行なわれる（ステップ30__4）。このようにして、処理実行中に、動的に、オブジェクトBからオブジェクトAのデータを直接に参照する経路が形成される。

【0163】

図31はオブジェクトBの処理中の部分処理の第10例を示すフローチャートである。ここでは、図31に示す部分処理に先立ってオブジェクトCが生成されている必要がある（図23参照）。オブジェクトCの生成タイミングは、この図31に示す部分処理の実行前であればいつでもあってもよい。

この図31に示す部分処理では、オブジェクトBに引数として渡された出力データタグテーブルが参照され、もともとはオブジェクトBのポインタ要素リスト番号であってオブジェクトCに引き継がれたポインタ要素リスト番号IN_B から、オブジェクトAのデータ要素リスト番号OUT_A が取得される（ステップ31__1）。次いで、その取得したデータ要素リスト番号OUT_A に対応する、オブジェクトAのデータ要素リストが生成され（ステップ31__2）、さらにポインタ要素リスト番号IN_B に対応する、オブジェクトCのポインタ要素リストが生成され（ステップ31__3）、それらオブジェクトAのデータ要素リストとオブジェクトCのポインタ要素リストとの結合処理が行なわれる（ステップ31__4）。

【0164】

このようにして、処理実行中に、いわば動的に、新たに生成されたオブジェクトCから、オブジェクトAのデータを直接に参照する経路が生成される。

尚、ここでは、オブジェクトBの処理中の各種の部分処理について説明したが、これら各種の部分処理は必ずしもそれぞれが単独で実行されるとは限らず、必要に応じて複数の部分処理が連続的に、あるいは組み合わされて実行される。

【0165】

図32は、図15に示すオブジェクト結合部20における入力インストラクションタグテーブル生成部24の処理を示すフローチャートである。

ここでは、コンピュータシステム100（図1参照）の操作者により、LSI間を配線で結ぶ要領で以下の指定が行なわれる。

- (a) オブジェクトAのポインタ
- (b) オブジェクトBのポインタ
- (c) オブジェクトAのメソッド番号ME_A
- (d) オブジェクトBのメッセージ番号MA_B

図32の処理では、上記の指定を受けて、オブジェクトBのメッセージの最大数MA_{BMAX}と同一幅の入力インストラクションタグテーブルの枠が生成され（ステップ32__1）、その枠のオブジェクトBのメッセージ番号MA_Bの欄にオブジェクトAのメソッド番号ME_Aが格納され（ステップ32__2）、オブジェクトAの、オブジェクトBに関する全てのメソッド要素（例えば図19に示すメソッド要素）に、その生成した入力インストラクションタグテーブルへのポインタが登録される（ステップ32__3）。図19には、入力インストラクションタグテーブルへのポインタ（P1）が既に登録された状態の図が示されている。

【0166】

ここでは代表的にオブジェクトBを取り上げているが、オブジェクトAの出力インストラクションバス部にはオブジェクトAから発行されたメッセージを受け取る可能性のある全てのオブジェクトに関する入力インストラクションタグテーブルが生成され、その入力インストラクションタグテーブルに対応するオブジェクトに関するメソッド要素全てに、その入力インストラクションタグテーブルへのポインタが登録される。以下に説明する出力インストラクションタグテーブル、入力データタグテーブル、および出力データタグテーブルの場合も同様である。

【0167】

図33は、図16に示すオブジェクト結合部30における出力インストラクションタグテーブル生成部34の処理を示すフローチャートである。

ここでは、コンピュータシステム 1 0 0（図 1 参照）の操作者により、L S I 間を配線で結ぶ要領で以下の指定が行なわれる。

- (a) オブジェクト A のポインタ
- (b) オブジェクト B のポインタ
- (c) オブジェクト A のメッセージ番号 MA_A
- (d) オブジェクト B のメソッド番号 ME_B

図 3 3 の処理では、上記の入力を受けて、オブジェクト B のメソッドの最大数 ME_{BMAX} と同一幅の出力インストラクションタグテーブルの枠が生成され（ステップ 3 3 __ 1）、その枠の、オブジェクト B のメソッド番号 ME_B の欄にオブジェクト A のメッセージ番号 MA_A が格納され（ステップ 3 3 __ 2）、さらに、オブジェクト A の、オブジェクト B に関する全てのメソッド要素に、その生成した出力インストラクションタグテーブルへのポインタが登録される（ステップ 3 3 __ 3）。図 1 9 は、出力インストラクションタグテーブルに関しても、既にその出力インストラクションタグテーブルへのポインタ（P 2）が登録された後の状態の図である。

【0 1 6 8】

図 3 4 は、図 1 7 に示すオブジェクト結合部 4 0 における入力データタグテーブル生成部 4 4 の処理を示すフローチャートである。

ここでは、コンピュータシステム 1 0 0（図 1 参照）の操作者により、L S I 間を配線で結ぶ要領で以下の指定が行なわれる。

- (a) オブジェクト A のポインタ
- (b) オブジェクト B のポインタ
- (c) オブジェクト A のポインタ要素リスト番号 IN_A
- (d) オブジェクト B のデータ要素リスト番号 OUT_B

図 3 4 に示す処理では、上記の指定を受けて、オブジェクト B のデータ要素リストの最大数 OUT_{BMAX} と同一幅の入力データタグテーブルの枠が生成され（ステップ 3 4 __ 1）、その枠の、オブジェクト B のデータ要素リスト番号 OUT_B の欄に、オブジェクト A のポインタ要素リスト番号 IN_A が格納され（ステップ 3 4 __ 2）、さらに、オブジェクト A の、オブジェクト B に関する全てのメソッ

ド要素に、その生成した入力データタグテーブルへのポインタが登録される（ステップ 3 4 __ 3）。図 1 9 は、入力データタグテーブルについても、その入力データタグテーブルへポインタ（P 3）が登録された後の状態の図である。

【0 1 6 9】

図 3 5 は、図 1 8 に示すオブジェクト結合部 5 0 における出力データタグテーブル生成部 5 4 の処理を示すフローチャートである。

ここでは、コンピュータシステム 1 0 0（図 1 参照）の操作者により、L S I 間を配線で結ぶ要領で以下の指定が行なわれる。

- （a）オブジェクト A のポインタ
- （b）オブジェクト B のポインタ
- （c）オブジェクト A のデータ要素リスト番号
- （c）オブジェクト B のポインタ要素リスト番号

図 3 5 の処理では、上記の入力を受けて、オブジェクト B のポインタ要素リストの最大数 IN_{BMAX} と同一幅の出力データタグテーブルの枠が生成され（ステップ 3 5 __ 1）、その枠の、オブジェクト B のポインタ要素リスト番号 IN_B の欄に、オブジェクト A のデータ要素リスト番号 OUT_B が格納され（ステップ 3 5 __ 2）、さらに、オブジェクト A の、オブジェクト B に関する全てのメソッド要素に、その生成した出力データタグテーブルへのポインタが登録される（ステップ 3 5 __ 3。図 1 9 は、出力データタグテーブルについても、その出力データタグテーブルへのポインタ（P 4）が登録された後の状態の図である。

【0 1 7 0】

尚、図 3 2 ～図 3 5 はそれぞれが単独に実行され、図 1 9 に示す 4 つのタグテーブルのうちの 1 つのみがメソッド要素に登録されてもよいが、必要に応じて、または常に、これら 4 つのタグテーブルのうち 2 つ以上あるいは全てが 1 つのメソッド要素に登録されてもよい。また、これら図 3 2 ～図 3 5 には明示的に示されていないが、これら図 3 2 ～図 3 5 に示す処理のうちのいずれかが実行された場合に、オブジェクト A 自身へのポインタ（P 5）もメソッド要素に登録される。

【0 1 7 1】

図 1 5 ～ 図 3 5 を参照して説明した実施形態によれば、オブジェクトどうしが初期状態として結びつけられるだけでなく、処理の実行中に動的に、メッセージとメソッドとの結合、データとポインタとの結合が行なわれる。新たなオブジェクトが生成されると、その新たなオブジェクトに関しても同様に、動的な結合が行なわれる。このように、そのときそのときの状況に応じて新たな結合が行なわれ、複数のオブジェクト間でのメッセージやデータの授受が極めて高速化される。

【0 1 7 2】

次に、図 2 に示すオブジェクトウェアプログラミングシステム 1 2 0 のオブジェクト間配線エディタ部 1 2 2 およびその周辺部に関連した実施形態について説明する。ここでは、オブジェクト間配線エディタ部 1 2 2 およびその周辺部に関連した実施形態のうち、図 1 に示すコンピュータシステム 1 0 0 の画像表示部 1 0 2 の表示画面 1 0 2 a への、オブジェクトと オブジェクト間の配線とからなるオブジェクト間ネットワークの表示形態について説明する。

【0 1 7 3】

前述したように、ソフトウェアの再利用性が低いことや実行速度が遅いことなど種々の欠点を持っていたものの、オブジェクト指向プログラミングという考え方が従来より存在し、オブジェクトどうしの接続関係を記述するためにオブジェクト間を配線するという考え方が存在していた。しかしながら、従来は、前述したように、データをメッセージの引数として他のオブジェクトに渡すなど、オブジェクト間の接続関係は極めて単純なものであった。ところが、前述したインタプリタ部 1 2 3 (図 2 参照) に関連した実施形態に示すようにオブジェクトどうしの配線というよりもさらに複雑に、オブジェクトの中のポインタどうしを配線する必要がある場合に、配線が複雑となり、従来のような表示方法では、オブジェクト間の接続関係を容易に理解し効率良く配線するのが難しいという問題がある。

【0 1 7 4】

例えば従来は、オブジェクト間ネットワークを表示する際は、オブジェクトの表示位置とオブジェクト間の配線を表示する位置に区別がなく、自由にオブジェ

クトの配置及び配線を行っている。このため、ある表示装置ではオブジェクトと配線が重なることを許しており、ユーザが重なりを回避しながら配線を行わなければならないという問題がある。また、ある表示装置では、配線をして、結果として出来上がるネットワークを表示しないため、オブジェクト間の関係を容易には理解できないという問題がある。

【0 1 7 5】

また、表示されているオブジェクトが階層構造になっており、あるオブジェクトを構成するサブネットワークを表示する装置においては、従来は、新たな画面（ウインドウ）にそのネットワークを表示している。そのため、親オブジェクトのネットワークとサブネットワークのつながりを確認することが困難であるという問題と、新たなウインドウにより親オブジェクトのネットワークが隠れてしまうという問題がある。

【0 1 7 6】

さらに、従来のオブジェクト間ネットワーク表示装置においては、オブジェクトの大きさは固定あるいは自由に変更できるようになっている。しかし、大きさが固定の場合では、オブジェクトの入出力端子の数が可変の場合、端子間隔が狭くなり端子名を表示することが困難になる。また、自由に変更できる場合でもユーザが調整をしなければならず、作業量が増すという問題がある。

【0 1 7 7】

さらに、従来のオブジェクト間ネットワーク表示装置においては、ネットワーク配線内のデータや指令の流れる方向を示す場合、線の終端にのみ、または両端にのみ矢印をつけるようにしている。従って線の途中では流れの方向を確認することが出来ず、線両端のオブジェクトが表示画面からはみ出している場合には、始端または終端となるオブジェクトの端子が入力端子であるか出力端子であるかの判断が出来ないという問題がある。

【0 1 7 8】

さらに、線が交差する場合、2つの線が交わっているのか分離しているのかを判断するために、交差点に黒い点などの印を付したり、円弧を利用している。しかし、黒い点などの印では接続状態を示す表示のルールをユーザが理解している

ことが必要となり、円弧では 1 本の線に対して半径分の幅が必要になるという問題がある。

【0179】

以下に説明する実施形態は、上述のオブジェクト間ネットワークの表示に伴う問題に鑑み、ユーザに理解しやすい表示方法を提供するものである。

以下、本発明のオブジェクト間ネットワーク表示方法の実施形態について説明する。ここでは、先ず本発明のオブジェクト間ネットワーク表示方法の基本的な実施形態についてひと通り説明し、次いでさらに具体的な実施形態について説明する。

【0180】

図 36 は、本発明のオブジェクト間ネットワーク表示方法の一実施形態の表示画面例を示す模式図である。尚、ここに示す表示画面 102 a は、図 1 に示すコンピュータシステム 100 に備えられた画像表示部 102 の表示画面 102 a であり、その表示画面 102 a 上に、この図では格子 201 が示されているが、この格子 201 は表示画面 102 a が複数の表示領域に区分けされていることを明示するためのものであり、表示画面 102 a には実際には表示されない。

【0181】

格子 201 で区切られた各升目 1 つずつからなる各表示領域には、オブジェクト指向プログラミングにより生成される複数のオブジェクトのうちのいずれか 1 つのオブジェクトを表示するオブジェクト表示部分領域 203 と、複数のオブジェクト間を結ぶ配線を表示する配線表示部分領域 204 が示されている。ここで配線とは、図 3、図 15～図 18 に示すメッセージの授受の経路 13 a, 23 a, 33 a, 43 a, 53 a や、図 3 に示すデータの授受の経路 16 a などであらわす線である。配線表示部分領域 204 は、隣接する 2 つの表示領域それぞれのオブジェクト表示部分領域 203 どうしの間にその配線表示部分領域 204 が形成されるようにその領域が定められている。

【0182】

ここでは、表示画面 102 a 上に、ネットワークを構成する複数のオブジェクトそれぞれが各表示領域内のオブジェクト表示部分領域 203 に 1 つずつ配置さ

れるとともに、これら複数のオブジェクトどうしを結ぶ配線が複数の表示領域に跨って連なる配線表示部分領域 204 に表示されてなる画像が表示される。

このような表示方法によれば、整然と並んだオブジェクトの配置が得られ、しかもオブジェクトを表示する領域と配線を表示する領域を分けたためオブジェクトと配線が重なることのない見やすい表示を得ることができる。

【0183】

次にネットワークを構成するオブジェクトが階層化されている場合のネットワークの表示方法について説明する。

図 37 は、階層化されたネットワークの説明図である。

この図 37 には、オブジェクト 1, 2 がそれぞれサブネットワーク 1, 2 で構成されている例が示されている。各サブネットワークは、それぞれが複数のオブジェクトと、それら複数のオブジェクトどうしを結ぶ配線とで構成されている。ここでは二段の階層構造が示されているが、三段以上の階層を持っていてもよい。

【0184】

図 38 は、多数のオブジェクトと配線からなる表示画像例を示す図であり、図 38 (A) は表示画像全体、図 38 (B) は、オブジェクト 205 を中心とした部分画像を示している。

また、図 39 は、図 38 に示すオブジェクト 205 に代えて、そのオブジェクト 205 を構成するサブネットワーク 206 を表示した表示画像例を示す図であり、図 39 (A) は表示画像全体、図 39 (B) はサブネットワーク 206 を中心とした部分画像を示している。

図 38 に示す表示画像 207 に含まれるオブジェクト 205 に代えて、上述したような下層の複数のオブジェクトとそれらのオブジェクト間の配線とで構成されたサブネットワークを表示するにあたり、このサブネットワーク 206 には、オブジェクト 205 の表示領域よりも広い表示領域が割り当てられ、このサブネットワーク 206 の表示領域の上下に並ぶ表示領域は左右に拡大され、このサブネットワーク 206 の表示領域の左右に並ぶ表示領域は上下に拡大させ、さらにサブネットワーク 206 の表示領域に対し斜めに位置する表示領域については、

そのサブネットワーク 2 0 6 と等価な要素であるオブジェクト 2 0 5 が表示された図 3 8 の表示画像中の表示領域と同一の寸法が割り当てられている。

【0 1 8 5】

図 3 9 (B) に示すように、サブネットワーク 2 0 6 を構成する複数のオブジェクト間には配線が表示されており、かつそのサブネットワーク 2 0 6 とそのサブネットワーク 2 0 6 の周囲のネットワークとの間の配線も表示されている。

このような表示方法を採用すると、従来例のように独立したウインドウにサブネットワーク 2 0 6 を表示する場合と比べ、サブネットワークとその周囲のオブジェクトのネットワークとの接続状態を容易に確認することが可能となる。

【0 1 8 6】

図 4 0 は、サブネットワークの表示方法の他の例を示す図であり、図 4 0 (A) はサブネットワークを表示する前の、そのサブネットワークと等価なオブジェクトを含む表示画像例、図 4 0 (B) はそのオブジェクトをサブネットワークに置き換えたときの表示画像例を示している。

図 4 0 (A) に示す表示画像 2 0 7 に含まれるオブジェクト 2 0 5 を、図 4 0 (B) に示すように、そのオブジェクト 2 0 5 と等価なサブネットワーク 2 0 6 に置き換えるものとする。

【0 1 8 7】

このとき、サブネットワーク 2 0 6 には、もとのオブジェクト 2 0 5 の表示領域よりも広い表示領域が割り当てられる。ただしオブジェクト 2 0 5 に代えてサブネットワーク 2 0 6 を表示した図 4 0 (B) に示す表示画像の周縁に位置する表示領域は図 4 0 (A) に示す表示画像の周縁に位置する表示領域と比べ同一のオブジェクトが表示された表示領域である。また、それら周縁に位置する表示領域の、表示画像 2 0 7 の外周に接する辺の位置および寸法は図 4 0 (A) と図 4 0 (B) とで同一である。すなわち、図 4 0 (A) と図 4 0 (B) は、オブジェクト 2 0 5 がサブネットワーク 2 0 6 に置きかわった点を除き、そのサブネットワーク 2 0 6 を除く表示領域は図 4 0 (B) ではゆがんではいるものの、図 4 0 (A) と同一の情報が表示されており、図 3 9 (A) と比べた図 3 9 (A) のときのように、オブジェクト 2 0 5 に代えてサブネットワーク 2 0 6 を表示したこ

とによってサブネットワーク 2 0 6 から離れた部分の表示領域が表示画面から外れてしまうことが防止される。

【0 1 8 8】

したがってこの表示方法を採用すると、図 3 9 の例と同様にサブネットワークとその周囲のオブジェクトのネットワークとの接続状態を容易に確認することができ、それに加えて、変形してはいるものの、サブネットワークを表示した状態で、サブネットワーク表示前（第 1 の画像）に表示されていたネットワーク全体を確認することができる。

【0 1 8 9】

図 4 1 は、複数の升目が連結された表示領域を有する表示画像例を示す図であり、図 4 1（A）は全体画像を示す図、図 4 1（B）、（C）は部分画像を拡大して示す図である。

この表示画像には各種の寸法のオブジェクト 2 1 0 ～ 2 1 5 が示されている。これらのオブジェクト 2 1 0 ～ 2 1 5 のうちオブジェクト 2 1 0 は、升目 1 つ分で区切られた表示領域内に配置されており、他の寸法のオブジェクト 2 1 1 ～ 2 1 5 は、隣接する複数の升目が連結されて 1 つの表示領域を構成した、拡大された表示領域内に配置されている。各オブジェクトは、図 4 1（B）、図 4 1（C）に示すように、各表示領域の形状および寸法に応じた形状および寸法に規格化されている。

【0 1 9 0】

このような表示方法を採用すると、種々の寸法のオブジェクトを見やすい寸法で表示できるとともに、規格化によってさらに見やすい表示画面が表示されることになる。

次にオブジェクト間をつなぐ配線の表示方法について説明する。

図 4 2 は、配線の表示方法に特徴を持つ表示画像例を示す図である。

【0 1 9 1】

表示画面 1 0 2 a にはオブジェクト 2 1 6 ～ 2 1 9 が表示されており、出力端子 2 2 0 と入力端子 2 2 1 が配線 2 2 2 で結ばれている。ここで、出力端子は、そのオブジェクトのデータあるいは指令（メッセージ）を他のオブジェクトに向

けて出力することを示しており、入力端子は、他のオブジェクトのデータあるいは指令（メッセージ）を受け取ることを示している。

【0192】

配線 222 は、出力端子 220 から入力端子 221 に向かう向きの情報を持った線であり、データあるいは指令流れの向きが、その配線を構成する短かい線分毎に繰り返し表示されている。

このような配線の表示方法を採用すると、その配線で結ばれるべきオブジェクトの一方もしくは双方が表示画面 102a からはみ出している場合であってもデータあるいは指令の流れる方向を一見して理解することができる。

【0193】

図 43 は、配線の表示方法の他の例を示す図である。

この配線 223 は、中心線 223a と、その中心線 223a の両脇に沿う縁線 223b とで構成されている。中心線 223a と縁線 223b は互いに異なる表示態様、例えば互いに異なる色相、明度、あるいは彩度の線である。

このような中心線 223a と縁線 223b とからなる配線 223 を採用し、2本の配線 223 が交差する場合において、それら 2本の配線が互いに異なるデータもしくは制御の流れを表わすものである場合は、図 43（A）に示すように一方の配線の中心線を他方の配線の縁線に接する位置もしくは近接する位置（この実施形態では接する位置）で分断して立体交差のように表示する。一方、それら 2本の配線が同一のデータもしくは制御の流れを表わす配線である場合は、図 43（B）に示すように、双方の配線の中心線 223a どうしが連続するように表示する。このような表示方法を採用すると、交差する配線が相互に接続されているのか、あるいは単に交差しているだけなのかを、容易に判断することができる。

【0194】

以上で、本発明のオブジェクト間ネットワーク表示方法の基本的な実施形態についての説明を終了し、以下、本発明のオブジェクト間ネットワーク表示方法の、さらに具体的な実施形態について説明する。

図 44 は、オブジェクトのネットワークを表示するための表示領域を作成する

手順の説明図である。図 4 4 (A) では表示画面の縦と横が 4 分割によって格子が作成されている。図 4 4 (B) では作成した格子のひと升に対し、縦横とも 50 % の長さで構成される領域をオブジェクトを配置する領域とし、これを升目の中央に位置するように構成されている。図 4 4 (B) では画面を等分割し、オブジェクトの領域を升目の中心に位置させたが、升目の幅を指定してもよく、図 4 4 (C) のようにオブジェクトの領域を升目の角に位置させてもよい。

【0 1 9 5】

図 4 5 は、ユーザにより表示画面上にオブジェクトが配置されているときの様子を示す図、図 4 6 はユーザにより、表示画面上に配置されたオブジェクト間の配線が行われているときの様子を示す図である。

図 4 5 に示すように、ユーザがオブジェクト 2 2 4 を配置するとき、本実施形態では、ユーザにより配置されたオブジェクト 2 2 4 は、その配置位置に一番近い、オブジェクト配置専用の領域 2 2 5 に、自動的に位置決めされる。従って、ユーザはオブジェクトを適当に配置するだけで整然と並べられたオブジェクトの配置が得られる。また、本実施形態では、図 4 6 (A) に示すように、ユーザはオブジェクト 2 2 6 とオブジェクト 2 2 7 の端子間を直線的に結ぶ操作 2 2 8 を行うだけで、図 4 6 (B) に示すように、ネットワークの配線を表示する領域 2 0 4 の中で自動的に配線 2 2 9 が形成される。従って、オブジェクトと配線が重なるようなことがなく、ユーザに見やすいネットワークを表示することが可能となる。

【0 1 9 6】

図 4 7 は、サブネットワーク表示前 (A) および表示後 (B) のオブジェクト間ネットワークの表示画面例を示す図、図 4 8 は、図 4 7 (A) の表示から図 4 7 (B) の表示への変更の手順を示すフローチャートである。

表示画面上に図 4 7 (A) の画像が表示されている段階で、例えば図示しないマウス等を操作してサブネットワークを有するオブジェクトを指定し (ステップ 4 8 __ 1)、さらにそのサブネットワークを表示するよう指定する (ステップ 4 8 __ 2)。すると画像表示装置内では、選択したオブジェクトの位置する升目とその上下左右に並ぶ格子を、選択したオブジェクトの升目の左上の角を起点とし

てサブネットワークの表示に必要な領域分縦横に拡大し（ステップ 4 8 __ 3）、この拡大に伴って、上下左右に並ぶオブジェクトの変形と配線の延長を行い（ステップ 4 8 __ 4）、サブネットワーク表示用に拡大した升目内に新たな格子を形成してそこにサブネットワークを表示し（ステップ 4 8 __ 5）、さらに、サブネットワークのオブジェクトと周囲のネットワークのオブジェクトとを接続する（ステップ 4 8 __ 6）。

【0 1 9 7】

このような手順により、図 4 7（A）に示す画像から図 4 7（B）に示す画像への変換が行われる。尚ここでは升目の拡大起点を升目の左上の角としているが、他の角であってもよく、あるいは升目の中心であってもよい。

図 4 9 は、サブネットワーク表示手順の説明図であり、図 4 9（A）はサブネットワーク表示前のオブジェクト間ネットワークを示した図、図 4 9（B）はサブネットワークの拡大表示と拡大した升の上下左右に台形の升目を作成した状態を示した図、図 4 9（C）はサブネットワークの拡大表示と周囲のオブジェクトの升目を変形してネットワーク全体を画面内に表示した状態を示した図である。

【0 1 9 8】

また、図 5 0 は、サブネットワーク表示手順を示すフローチャートである。

図 5 0 に示すように、例えばマウス等の操作子を操作して、サブネットワークを有するオブジェクトを選択し（ステップ 5 0 __ 1）、そのサブジェクトに対応するサブネットワークを表示するよう指定する（ステップ 5 0 __ 2）。すると画像表示装置内では、以下の手順で、図 4 9（A）に示す画像から図 4 9（C）に示す画像への変換が行われる。

【0 1 9 9】

先ず、ステップ 5 0 __ 3 においてサブネットワークが表示画面に収まるか否かが判定され、収まらないときは図 4 9（A）の画像から図 4 9（B）の画像への変換は行わない。サブネットワークが表示画面内に収まると判定されると、ステップ 5 0 __ 4 に進み、選択したオブジェクトの位置する升目の中心（図 4 9（A）参照）を起点として、その升目をサブネットワークの表示に必要な領域分、縦横に拡大する。

【0200】

次いで、図49（B）に示すように、拡大された升目の角から升目の上下左右の画面端の升目の角へ直線を引き、台形を作る（ステップ50__5）。さらに、その台形を必要な分割数で分割し、台形の升目を作成する（ステップ50__6）。次いで、台形の升目の角から画面端の升目の角へ直線を引き、残りの升目を作成する（ステップ50__7）。さらに升目が変形されたことに伴うオブジェクトの変形と配線を行う（ステップ50__8）。最後に、サブネットワークのオブジェクトと周囲のネットワークのオブジェクトを接続する（ステップ50__9）。

【0201】

このような手順により、図49（A）に示す画像から図49（C）に示す画像への変換が行われる。

尚、ここではサブネットワークの上下左右の升目を台形にしているが、曲線を使った升目にしてもよい。

図51は、オブジェクトを表示する表示領域を1つもしくは複数の連結された升目で形成した例を示す模式図である。ここでは、オブジェクトの端子の数によって使用する升目の数を変更している。図51（A）は1升目を1つ使用した場合であって最大12端子、図51（B）は2升目を2つ使用した場合で最大30端子、図51（C）は4升目を4つを使用した場合であって最大48端子のオブジェクトを表現している。端子数のより多いオブジェクトに対しては使用する升目の数を増やして対応することができる。

【0202】

図52は、配線中（A）および配線後（B）の配線の表示方法の例を示す模式図である。

図52（A）中、画面1はオブジェクト1（obj1）の出力端子からオブジェクト2（obj2）の入力端子へ配線をしている途中を示している。画面1からはオブジェクト1が見えなくなっているが、引出した線の形状から、接続すべき端子は入力端子であることが判断できる。画面2についても同様に、オブジェクト4（obj4）の入力端子からオブジェクト3（obj3）の出力端子に配線する場合、画面2からオブジェクト4が見えなくなっても、線の形状から、接

続すべき端子は出力端子であることがわかる。図 5 2 (B) は配線終了後のネットワークを示しており、配線は通常の実線に変更されている。ここでは、配線が全て終了した時点で通常の実線に変更しているが、1つの配線が終了する毎に変更してもよい。

【0 2 0 3】

図 5 3 は配線の表示例を示す模式図、図 5 4 は、図 5 3 に示す配線を描く手順を示すフローチャートである。

ここでは、図 4 3 を参照して説明した、中心線とを緑線とからなる配線が採用されており、ユーザが接続すべき出力端子と入力端子を選択すると、図 5 4 に示す手順で自動配線が行なわれる。

【0 2 0 4】

ユーザが、接続すべき出力端子と入力端子を選択すると（ステップ 5 4 __ 1）、先ず出力端子側に縦レーン A を作成し（ステップ 5 4 __ 2）、出力端子から縦レーン A まで出力端子の横レーンを線で上書きする（ステップ 5 4 __ 3）。この上書きにより、表示画面上には、上書きされた部分について配線が表示される。ステップ 5 4 __ 4 では、入力端子と出力端子が対向するか否かが判定される。入力端子と出力端子が対向するとは、例えば出力端子 1 と入力端子 q のように、出力端子と入力端子が互いに向き合った辺に存在することを意味し、例えば出力端子 7 と入力端子 r の場合は対向していないものと判断される。

【0 2 0 5】

入力端子と出力端子が対向していると判断された場合は、ステップ 5 4 __ 5 に進み、入力端子の横レーンまで、縦レーン A に線を上書きする。例えば出力端子 8 と入力端子 C との間の配線が既に行なわれ、今回出力端子 8 と入力端子 e との間の配線を行なう場合のように、既に横レーンと繋がっている部分があれば、図 4 3 (B) に示すような結合処理を行ない（ステップ 5 4 __ 5）、入力端子の横レーンに、入力端子まで線を上書きする（ステップ 5 4 __ 6）。

【0 2 0 6】

ステップ 5 4 __ 4 において入力端子と出力端子が対向していないと判定されると、ステップ 5 4 __ 7 に進んで、入力端子側に縦レーン B を作成し、さらにステ

ップ54__8においてオブジェクトに挟まれていない横レーンCを作成し、さらにステップ54__9において横レーンCまで縦レーンAに線を上書きする。ここで、既に横レーンと繋がっている部分があれば結合処理を行なう。

【0207】

さらにステップ54__10に進んで横レーンCを縦レーンBまで線を上書きし、ステップ54__11において入力端子の横レーンまで縦レーンBに線を上書きする。既に横レーンと繋がっている部分があれば結合処理を行なう。

その後ステップ54__6に進んで、入力端子の横レーンに、入力端子まで線を上書きする。

【0208】

図55～図57は、配線手順の他の例を示すフローチャート、図58～図62、および図63は図55～図57に示す配線手順に従って配線を行なった結果を示す模式図である。ここに示す配線手順を採用すると、図54に示す配線手順を採用した場合と異なり、不揃いの形状のオブジェクトが存在していても、それを回避して自動配線することができる。

【0209】

図55に示すように、接続すべき出力端子と入力端子を選択すると（ステップ55__1）、出力端子のある配線領域に、出力端子に直角なレーン1（図58～図62参照）を設け（ステップ55__2）、出力端子のレーンであるレーン2（図58～図61参照）上を、出力端子からレーン1まで線を引く（ステップ55__3）。次に、出力端子と入力端子が対向するか否か判定される（ステップ55__4）。図58～図61のうちの図58に示すように出力端子と入力端子が対向する場合、ステップ55__5に進み、レーン1とレーン2との交点aから入力端子のレーンであるレーン3まで線を引き、さらにレーン1とレーン3との交点である交点bから入力端子まで線を引く（ステップ55__6）。これにより、図58のように出力端子を入力端子が対向する場合の配線が完了する。

【0210】

ステップ55__4において出力端子と入力端子が対向していないと判定された場合、ステップ55__7に進み、レーン1とレーン2との交点aから、レーン1

上を、入力端子のあるオブジェクトの方向へ線を引く。この線を引きながら、その線が既存のオブジェクトとぶつかったか否か（図56ステップ56__1）、その線が入力端子を持つオブジェクトの配線領域に到達したか否か（ステップ56__2）、その線が入力端子の配線領域に到達したか否か（ステップ56__3）、およびその線が入力端子のレーンであるレーン3と直角の位置まで到達したか否か（ステップ56__4）が判定される。

【0211】

ステップ56__1において既存のオブジェクトにぶつかったことが判定されると、ステップ56__10に進み、図63（A）に示すように、その線の現在の先端が存在する位置の配線領域に線に直角なレーンAを設けてその線と接続し、さらに、入力端子に近い配線領域にその線に平行のレーンBを設けて、レーン1からレーンBまで、レーンAに沿って線をつなぎ（ステップ56__11）、レーンAとレーンBとの交点kから入力端子のあるオブジェクトの方向へ、レーンBに沿って線を引く（56__13）。

【0212】

ステップ56__2は、入力端子を持つオブジェクトの領域（オブジェクト自身の配置領域だけでなくそのまわりの配線領域も含む。例えば図58における、入力端子を持つオブジェクトの領域とは、部分領域p, q, r, s, u, v, w, xの全てを指す）を上下および左右に延ばした十字の領域内まで、レーン1に沿って線を引いた段階で判定する。このステップ56__2において、その線が入力端子を持つオブジェクトの配線領域（図58の例では部分領域p, q, r, s, u, v, w, xを指す）に到達していないときは、ステップ56__12に進み、図63（C）に示すように、線の先端が現在存在する位置の配線領域に、その線に直角なレーンCを設けて線をつなぎ、ステップ56__13において、その交点kからレーンCに沿って入力端子のあるオブジェクトの方向へ線を引く。

【0213】

ステップ56__3において線が入力端子の配線領域（図58に示す例では部分領域p, s, vが、入力端子の配線領域に相当する）に到達していないと判定されたときは、図57のステップ57__1に進む。この場合については後述する。

ステップ56__4において線と入力端子のレーン3とが直角か否か判定され、直角の場合、図59に示すように、線をレーン3まで延長し（ステップ56__5）、レーン3との交点Cからレーン3上を入力端子まで線を引く（ステップ56__6）。以上により、例えば図59に示す配線が完了する。

【0214】

一方、ステップ56__4において、線と入力端子のレーン3とが直角ではないと判定された場合、図60に示すように、入力端子の配線領域に、線に直角なレーン4を設け（ステップ56__7）、交点dからレーン3まで線を引き（ステップ56__8）、交点eから入力端子まで線を引く（ステップ56__9）。以上により例えば、図60に示す配線が完成する。

【0215】

ステップ56__3において線が入力端子の配線領域に到達しないと判定されると、図57のステップ57__1に進み、線と入力端子のレーン3とが直角か否か判定される。直角であると判定されると、図61に示すように、現在の配線領域にレーン5を設け（ステップ57__2）、入力端子の配線領域にレーン6を設け（ステップ57__3）、交点fからレーン5に沿ってレーン6まで線を引き（ステップ57__4）、交点gからレーン3まで線を引き（57__5）、交点hから入力端子まで線を引く（ステップ57__6）。これにより、例えば図61に示すような配線が完成する。

【0216】

ステップ57__1において線と入力端子のレーン3が直角ではないと判定されると、図62に示すように、入力端子の配線領域に直に直角なレーン7を設け（ステップ57__7）、線を交点aからレーン7まで延長し（ステップ57__8）、交点iからレーン3まで線を引き（ステップ57__9）、現在の配線領域に線に直角なレーンを設けて線と接続し（ステップ57__10）、交点jから入力端子に線を引く（ステップ57__11）。これにより、例えば図62に示すような配線が完成する。

【0217】

図55～図57の配線手順を踏むと、図63（A）～（D）の各オブジェクト

配置状態の場合も、そこに示す配線が完成する。

以上説明したとおり、本発明のオブジェクト間ネットワーク表示表法の実施形態によれば、理解しやすいオブジェクト間のネットワークを表示することができ、ユーザの作業能率の向上に大きく寄与することができる。

【0 2 1 8】

以上で、図 2 に示すオブジェクトウェアプログラミングシステム 1 2 0 のオブジェクト間配線エディタ部 1 2 2 およびその周辺部に関連した実施形態のうち、図 1 に示すコンピュータシステム 1 0 0 の画像表示部 1 0 2 の表示画面 1 0 2 a へのオブジェクト間ネットワークの表示方法に関する実施形態の説明は終了し、次いで、そのオブジェクト間配線エディタ部 1 2 2 およびその周辺部における、プログラミングに関する実施形態について説明する。このオブジェクト間配線エディタ部 1 2 2 におけるプログラミングは、表示画面上に、これまで説明したようなオブジェクト間ネットワークを表示しておき、その表示を見ながら、オブジェクト間を「配線する」ことにより行なわれる。

【0 2 1 9】

前述したように、ソフトウェアの再利用性や実行速度に問題を残しつつも、従来よりオブジェクト指向プログラミングという考え方が存在し、オブジェクトどうしの接続関係を記述するために、表示画面上にオブジェクトを模式的に表示しそれらのオブジェクト間を「配線する」という考え方が存在していた。そのような「配線」においては、以下のような問題がある。

【0 2 2 0】

オブジェクトが階層化されている場合において、異なる階層間のオブジェクトを直接接続することができず下層のオブジェクト（これを「子オブジェクト」と称する）を包含する一段上層のオブジェクト（これを「親オブジェクト」と称する）を経由して同一の階層内でのみ配線を行なうことができるようにした方式の場合、接続しようとするオブジェクトが離れた階層にあったときに、接続のために中継用の端子を多数作ることになり、配線に多くの手順を必要とし煩わしい。

【0 2 2 1】

一方、オブジェクトが階層化されている場合において、異なる階層内のオブジェクトを直接に接続することを可能にした方式の場合、階層を無視した配線図となり、見にくく、全体としての配線構造を把握しにくくなってしまうという問題がある。

また、一旦配線したオブジェクトを他のオブジェクトに置き換える必要を生じたとき、この置き換えを実現するためには、置き換え前のオブジェクトの配線を外し、置き換え後のオブジェクトに対しあらためて配線し直す必要があり、置き換えのために多くの手順を必要とすることになる。

【 0 2 2 2 】

ある階層に一旦配置して配線したオブジェクトを他の階層、例えば一段下の階層に移動させる場合も同様であり、移動前のオブジェクトの配線を外し、そこに親オブジェクトを配置して配線し、移動したオブジェクトをその親オブジェクトの子オブジェクトとして配置し、その親オブジェクトと子オブジェクトとの間を配線するといった多くの手順が必要となる。

【 0 2 2 3 】

さらに、従来の方式では、オブジェクト間配線が複雑になるにつれ、オブジェクト間の接続関係が配線図表示からは理解しにくくなるという問題がある。特に、処理の依頼の流れをあらわすバス（このバスを「インストラクションバス」と称する）が複数のオブジェクトに分岐して接続されているような場合に、それら複数のオブジェクトどうしでの処理の実行順序を配線図表示から把握することは困難であり、したがってその実行順序をその配線図上で変更することも困難である。

【 0 2 2 4 】

以下に説明する実施形態は、上述のオブジェクト間の配線に伴う問題に鑑み、配線作業を容易にするための手法に関する実施形態である。

図 6 4 は、本発明のオブジェクト指向プログラミング支援装置およびオブジェクト指向プログラミング用プログラム記憶媒体の基本構成図である。

オブジェクト指向プログラミング支援装置 3 0 0 は、データと処理とを有するオブジェクト複数を指示に応じて相互に結合するオブジェクト指向プログラミン

グを支援する装置であって、表示手段 3 0 1 と、オブジェクト結合手段 3 0 2 と、階層構造構築手段 3 0 3 と、操作子 3 0 4 とを備えている。

【0 2 2 5】

表示手段 3 0 1 は、オブジェクト本体をあらわすブロックと、自オブジェクト内部のデータを他オブジェクトに渡すためのデータ出力端子と、他オブジェクトのデータを自オブジェクトが受け取るためのデータ入力端子と、他オブジェクトに向けて処理を依頼するメッセージを発行するためのメッセージ端子と、他オブジェクトからの処理依頼を受け取ってメソッドを実行するためのメソッド端子とで表現されたオブジェクトであって、1つのオブジェクトの内部に1つもしくは複数のオブジェクトの存在を許容する階層構造で表現されたオブジェクトを表示するとともに、複数のオブジェクトの端子間を結ぶ配線を表示する手段であり、図 1 に示すコンピュータシステム 1 0 0 では、画像表示部 1 0 2 と、その画像表示部 1 0 2 の表示画面 1 0 2 a 上に上記のオブジェクトや配線を表示するためのソフトウェアと、そのソフトウェアを実行する C P U 等によりこの表示手段 3 0 1 が構成されている。

【0 2 2 6】

また、オブジェクト結合手段 3 0 2 は、複数のオブジェクトの端子間を配線で結ぶ指示に応じて、複数のオブジェクト間の、指示に応じた配線に対応した結合構造を構築する手段であり、図 1 に示すコンピュータシステム 1 0 0 では、この結合構造を構築するためのソフトウェアと、そのソフトウェアを実行する C P U 等により構成されている。

【0 2 2 7】

また、階層構造構築手段 3 0 3 は、オブジェクトの階層構造を構築する手段であり、図 1 に示すコンピュータシステム 1 0 0 では、この階層構造を構築するためのソフトウェアと、そのソフトウェアを実行する C P U 等により構成されている。

さらに、操作子 3 0 4 は、操作者（ユーザ）による操作に応じて、オブジェクト結合手段 3 0 2 に対しオブジェクト間を結ぶ配線を指示入力するとともに、階層構造構築手段 3 0 3 に対し、オブジェクトの、階層構造上の配置位置を指示入

力する操作子であり、図 1 に示すコンピュータシステム 100 では、キーボード 103 やマウス 104 およびそれらキーボード 103 やマウス 104 の操作をコンピュータシステム内部に取り込むためのソフトウェア等により構成されている。

【0228】

なお、ここではオブジェクト結合手段 302 を実現するためのソフトウェア自体もオブジェクト結合手段と称し、これと同様に階層構造構築手段 303 を実現するためのソフトウェア自体も階層構造構築手段と称し、これらソフトウェアとしてのオブジェクト結合手段 302 と階層構造構築手段 303 とを合わせたプログラムが本発明にいうオブジェクト指向プログラミング用プログラムに相当し、このオブジェクト指向プログラミング用プログラムが格納された記憶媒体 310 が、本発明にいうオブジェクト指向プログラミング用プログラム記憶媒体に相当する。図 1 に示すコンピュータシステム 100 では、このようなオブジェクト指向プログラミング用プログラムが記憶された状態の記憶部 105 が本発明にいうオブジェクト指向プログラミング用プログラム記憶媒体の一形態に相当し、そのオブジェクト指向プログラミング用プログラムが MO110 に格納されているときは、そのオブジェクト指向プログラム用プログラムが格納された状態にある MO110 も、本発明にいうオブジェクト指向プログラミング用プログラム記憶媒体に相当する。

【0229】

図 65 は、オブジェクトの内包関係の一例を示す概念図、図 66 は、オブジェクトどうしの、階層構造を規定する接続関係を示す模式図である。

図 65 に示すように、全体を 1 つのオブジェクトと考え、これをオブジェクト A とする。このオブジェクト A の中には、3 つのオブジェクト、すなわち、オブジェクト B、オブジェクト C、およびオブジェクト D が存在する。さらに、オブジェクト C の中には、オブジェクト E、オブジェクト F、およびオブジェクト G が存在する。さらに、オブジェクト F の中には、オブジェクト H が存在する。

【0230】

これを階層構造であらわすと、図 66 に示すようにあらわされる。このように

あらわされたオブジェクトの階層構造を、ここでは「オブジェクトツリー」と称する。

図66において水平方向に並ぶオブジェクトは同一の階層に配置されていることを意味し、上下方向に接続されたオブジェクトは、上の階層に配置されたオブジェクトが親オブジェクト、下の階層に配置されたオブジェクトが、その親オブジェクトに対する子オブジェクトであることを意味している。

【0231】

図67は、1つのオブジェクトの、他のオブジェクトへの接続関係を定めるポインタを示した模式図である。

各オブジェクトには、親子関係を規定するポインタとして、‘上階層へのポインタ’と‘下階層へのポインタ’とからなる‘上下階層へのポインタ’が存在し、また、同一の階層にあるオブジェクトどうしを接続するポインタとして‘FROM’と‘TO’との2つのポインタからなる‘同階層へのポインタ’が存在する。また、オブジェクト間のデータやインストラクションの流れをあらわす配線用のポインタとして、‘IN’と‘OUT’との2つのポインタからなる‘バスへのポインタ’と、‘インストラクション’と、‘データ’と、‘タグインストラクション’と、‘タグデータ’との4つのポインタからなる‘ケーブルへのポインタ’が存在する。

【0232】

上下階層へのポインタを構成する2つのポインタである上階層へのポインタおよび下階層へのポインタは、例えば図66に示すオブジェクトAの場合、それぞれ配線エディタへのポインタおよびオブジェクトBへのポインタである。

また、同階層へのポインタを構成する2つのポインタである‘FROM’と‘TO’は、例えば図66に示すオブジェクトCの場合、それぞれ、オブジェクトBへのポインタおよびオブジェクトDへのポインタである。

【0233】

このようにして、上下階層へのポインタと同階層へのポインタにより、例えば図66に示すような階層構造が構成される。

図68は、図67に示す‘バスへのポインタ’に接続されるバス要素リストを

構成する1つのバス要素を示す模式図、図69は、図67に示す‘ケーブルへのポインタ’に接続されるケーブル要素リストを構成するケーブル要素を示す模式図である。図70は、オブジェクトの間の配線例を示す模式図である。

【0234】

バス要素リストに並ぶ各バス要素は、そのオブジェクトの、他オブジェクトと接続されるバス（端子）を定義するものであり、ケーブル要素リストに並ぶ各ケーブル要素は、そのオブジェクトとを親オブジェクトとしたときの子オブジェクトどうしの端子どうしの結合関係（配線）を定義するものである。

図67には、バスへのポインタを構成するポインタとして‘IN’と‘OUT’との2つのポインタが示されているが、このうち‘IN’には、他オブジェクトからこの図67に示すオブジェクトにデータもしくはメッセージを入力するバスを定義するバス要素リストが接続され、‘OUT’には、この図67に示すオブジェクトから他オブジェクトに向けてデータもしくはメッセージを出力するバスを定義するバス要素リストが接続される。

【0235】

図67には、バスへのポインタを構成する2つのポインタ‘IN’と‘OUT’のうち‘IN’に2つのバス要素BUS1，BUS2からなるバス要素リストが接続されている。具体的には‘IN’にはバス要素BUS1が接続され、バス要素BUS1にはバス要素BUS2が接続されている。また‘OUT’には、2つのバス要素BUS3，BUS4からなるバス要素リストが接続されている。具体的には、‘OUT’に、バス要素BUS3が接続され、バス要素BUS3にはバス要素BUS4が接続されている。

【0236】

各バス要素は、図68に示すように、‘本体オブジェクトへのポインタ’と、‘本体オブジェクトのバスへのポインタ’と、‘次のバス要素（BUS）へのポインタ’と、‘その他のデータ’とから構成されている。尚ここでは、オブジェクトの端子を‘バス’と称している。

図70に示す構造の場合であって、図67に示すオブジェクトが図70のオブジェクトAである場合に、バス要素BUS1は、図70に示すオブジェクトAの

、例えば‘バス1’に対応し、図68に示すバス要素のうちの‘本体オブジェクトへのポインタ’は、図70に示すオブジェクトAに内包されるオブジェクトBとオブジェクトCのうち、オブジェクトAのバス1と接続されるオブジェクト（ここではオブジェクトB）へのポインタである。また、バス要素BUS1における‘本体オブジェクトのバスへのポインタ’は、本体オブジェクトであるオブジェクトBの、オブジェクトAのバス1と接続されるバス（図70の場合、オブジェクトBのバス1）へのポインタである。また、バス要素BUS1を構成する‘次のBUS（バス要素）へのポインタ’は、図67における要素BUS1の場合、バス要素BUS2へのポインタである。さらにそのバス要素BUS1を構成する‘その他のデータ’には、そのバス要素に対応するバス（ここに示す例では図10に示すオブジェクトAのバス1）が、データを授受するためのバスであるか、メッセージ（インストラクション）を授受するためのバスであるかの区別が含まれている。尚、前述したように、データもしくはインストラクションを受け取る側のバス（IN）であるか、データもしくはインストラクションを出力する側のバス（OUT）であるかは、図67に示すように、‘バスへのポインタ’が‘IN’と‘OUT’とに分かれていることで区別されている。

【0237】

図67には、‘ケーブルへのポインタ’を構成する4つのポインタ、すなわち、‘インストラクション’、‘データ’、‘タグインストラクション’、‘タグデータ’が存在し、それぞれにケーブル要素リストが接続される。

ただし、図67では、図示の煩雑さを避けるため、‘データ’に接続されたケーブル要素リストのみが示されており、‘データ’には直接的にはケーブル要素CABLE1が接続され、そのケーブル要素CABLE1にはケーブル要素CABLE2が接続され、そのケーブル要素CABLE2にはケーブル要素CABLE3が接続されている。

【0238】

‘ケーブルへのポインタ’は、親オブジェクトが自分の子オブジェクトどうしのバスの接続状態（配線）を管理するものであり、図70に示す例では、オブジェクトBとオブジェクトCとの間のバスどうしの配線を管理している。尚、親オ

ブジェクトであるオブジェクトAと、子オブジェクトであるオブジェクトBとの間、もしくは親オブジェクトAと子オブジェクトであるオブジェクトCとの間の配線は、上述したように、‘バスへのポインタ’に接続されたバス要素リストが管理している。

【0239】

ケーブルへのポインタを構成する4つのポインタ‘インストラクション’，‘データ’，‘タグインストラクション’，‘タグデータ’は、それぞれ、メッセージ（インストラクション）の流れを表わす配線、データの流れをあらわす配線、前述した、実行中に動的に形成されるインストラクションの流れをあらわす配線、および、実行中に動的に形成されるデータの流れをあらわす配線を管理している。

【0240】

図69に示すように1つのケーブル要素CABLEには、2つのターミナル要素TERMINALが付属している。ケーブル要素CABLEは、2つのターミナル要素TERMINALのうちの1番目のターミナル要素へのポインタと、次のケーブル要素へのポインタとから構成されている。またターミナル要素TERMINALは、‘オブジェクトへのポインタ’と、‘そのオブジェクトのバスへのポインタ’と、‘次のターミナル要素へのポインタ’とから構成されている。

【0241】

図69は、図70に示す、オブジェクトBのバス2とオブジェクトCのバス1とを結ぶ配線を管理するケーブル要素であり、1番目のターミナル要素には、オブジェクトBへのポインタと、オブジェクトBのバス2へのポインタとが格納され、2番目のターミナル要素には、オブジェクトCへのポインタと、そのオブジェクトCのバス1へのポインタとが格納されている。これにより、図70に示すオブジェクトBのバス2とオブジェクトCのバス1とが配線されていることになる。尚、2つのターミナル要素のうちの1番目のターミナル要素は、データないしインストラクションの出力側のバスに対応し、2番目のターミナル要素は、データないしインストラクションの入力側のバスに対応している。

【0242】

この図 6 9 に示すケーブル要素は、前述したように、オブジェクト B とオブジェクト C との共通の親オブジェクトであるオブジェクト A が管理している。

以上で、オブジェクトの階層構造を決めるためのポインタの管理、オブジェクトのバスを決めるためのポインタの管理、およびオブジェクトのバスどうしを接続する配線を決めるためのポインタの管理の一般的な説明を終え、次に、本発明のオブジェクト指向プログラミング支援装置および本発明のオブジェクト指向プログラミング用プログラム記憶媒体に格納されたオブジェクト指向プログラミング用プログラムの具体的な実施形態について説明する。

【0243】

本発明のオブジェクト指向プログラミング支援装置のうちの第 1 のオブジェクト指向プログラミング支援装置、および本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第 1 のオブジェクト指向プログラミング用プログラム記憶媒体に格納されたオブジェクト指向プログラミング用プログラムは、図 6 4 に示す階層構造構築手段 3 0 3 が、操作子 3 0 4 からの指示に応じて、指定されたオブジェクトである本体オブジェクトを複写して分身オブジェクトを作成し、その分身オブジェクトを、その本体オブジェクトが配置され階層とは異なる階層に配置する手段を有し、オブジェクト結合手段 3 0 2 が、分身オブジェクトが配置された階層における配線において、操作子 3 0 4 からの、分身オブジェクトと他オブジェクトとの間の配線の指示を受け付けるとともに、その分身オブジェクトと、その分身オブジェクトに対応する本体オブジェクトとを 1 つのオブジェクトとした結合構造を構築するものである。

【0244】

図 7 1 は分身オブジェクトの概念図、図 7 2 は、図 7 1 に示すオブジェクトの階層構造（オブジェクトツリー）を示す模式図である。

配線エディタにオブジェクト A が接続されており、そのオブジェクト A の下階層にオブジェクト B が接続され、オブジェクト B には、同階層にオブジェクト C が接続され、オブジェクト C には、その下階層にオブジェクト D が接続され、オブジェクト D には、その同階層にオブジェクト E が接続されている。

【0245】

ここで、異なる階層に配置されているオブジェクトBとオブジェクトEとを配線で接続する場合に、オブジェクトEの親オブジェクトであるオブジェクトCに接続のためのバス（端子）を形成し、そのオブジェクトCの端子と、オブジェクトEのバスとを接続し、さらにそのオブジェクトCの端子とオブジェクトBの端子とを接続してもよいが、これでは配線に手間がかかるため、ここでは、オブジェクトBやオブジェクトCが配置された階層に、それとは別の階層に配置されたオブジェクトEを本体オブジェクトとする分身オブジェクトE' が配置され、オブジェクトBや分身オブジェクトE' が配置された階層上で、オブジェクトBのバスと分身オブジェクトE' のバスとが配線される。

【0246】

図73は、分身オブジェクトの作成処理を示すフローチャートである。

まず、指定されたオブジェクト（ここではオブジェクトE）について、オブジェクトを複写して分身オブジェクトEを作成する（ステップ73__1）。ただし、ここではオブジェクト間の配線を目的としており、オブジェクトEを構成するプログラムの実体までする必要はなく、オブジェクトの表示および配線に必要な情報のみが複写される。このようにして、本発明にいう「複写」は、オブジェクトの表示および配線に必要な情報の複写を意味する。

【0247】

次に、オブジェクトEの全てのバスについて、

1. 分身オブジェクトE' にコピーバス（コピーバス要素）を作成し、
2. 作成されコピーバス要素に、本体オブジェクトEへのポインタ、および本体オブジェクトEの対応するバスへのポインタを書き込む（ステップ73__2）

図74は、本体オブジェクトと分身オブジェクトとの接続関係を示す模式図である。

【0248】

分身オブジェクトE' には、本体オブジェクトEの‘バスへのポインタ’に並ぶバス要素BUS1, BUS2, …がそのままの配列でコピーされ、そのコピーされた分身オブジェクトE' のバス要素BUS1, BUS2, …には、それぞれ

、本体オブジェクト E へのポインタと、本体オブジェクト E の、対応するバスへのポインタが格納される（図 6 8 参照）。このようにして分身オブジェクトを作成した後、図 7 1 に示すようにオブジェクト B と分身オブジェクト E' との間の配線が指示されると、この配線は、オブジェクト B と分身オブジェクト E' の親オブジェクトであるオブジェクト A の‘ケーブルへのポインタ’の先にその配線に対応するケーブル要素および 2 つのターミナル要素が配列される（図 6 9 参照）。

【 0 2 4 9 】

配線作業が終了し、図 2 に示すインタプリタ用配線データファイルに格納するインタプリタ用配線データを生成する際は、分身オブジェクト E' のバス要素リストから本体オブジェクト E の対応するバス要素がを見つけ出され、分身オブジェクト E' と本体オブジェクト E とを 1 つのオブジェクトとしたオブジェクト間の結合構造が構築される。

【 0 2 5 0 】

次に、本発明のオブジェクト指向プログラミング支援装置のうちの第 2 のオブジェクト指向プログラミング支援装置、および本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第 2 のオブジェクト指向プログラミング用プログラム記憶媒体に格納されたオブジェクト指向プログラミング用プログラムの実施形態について説明する。

本発明のオブジェクト指向プログラミング支援装置のうちの第 2 のオブジェクト指向プログラミング支援装置、および本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第 2 のオブジェクト指向プログラミング用プログラム記憶媒体に格納されたオブジェクト指向プログラミング用プログラムは、図 6 4 に示すオブジェクト結合手段 3 0 2 が、操作子 3 0 4 からの、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトの、他オブジェクトとの結合構造を解除して、置き換え前のオブジェクトの他オブジェクトとの結合構造を、置き換え後のオブジェクトに継承させるものであり、階層構造構築手段 3 0 3 が、操作子 3 0 4 からの、オブジェクトの置き換えの指示に応じて、置き換え前のオブジェクトが配置されていた階層に、その置き換え前のオブジェクトに

代えて、置き換え後のオブジェクトを配置するものである。

【0251】

図75は、オブジェクトの置き換え前のオブジェクトの結合関係を示す概念図、図76は、図75に示すオブジェクトに関するオブジェクトツリーを示す模式図である。

配線エディタにオブジェクトAが接続されており、そのオブジェクトAの下階層にオブジェクトBが接続され、オブジェクトBには同階層にオブジェクトCが接続され、オブジェクトCには、やはり同階層にオブジェクトDが接続されている。また、この階層構造に組み込まれていないオブジェクトEが存在している。ここで、オブジェクトCをオブジェクトEに置き換えるものとする。

【0252】

図77は、オブジェクト置き換え後のオブジェクトの結合関係を示す概念図、図78は、オブジェクト置き換え後のオブジェクトツリーの一部を示す模式図である。

オブジェクトCがオブジェクトEに置き換えられると、置き換え前のオブジェクトCの配線がそのままオブジェクトEに引き継がれ、階層構造上も、オブジェクトCが配置されていた階層に、そのオブジェクトCに代えて、オブジェクトEが配置される。

【0253】

図79は、オブジェクト置き換え処理を示すフローチャートである。

表示画面102a（図1参照）に図75に示すようなオブジェクト間ネットワークを表示しておき、マウス104を操作して置き換え後のオブジェクト（ここではオブジェクトE）をドラッグ（表示画面101aに表示されたオブジェクトEにマウスカーソルを合わせてマウスボタンを押し、そのままマウスを動かす操作をいう）してオブジェクトCに重ねる。オブジェクトEがドラッグされると、図64に示すオブジェクト結合手段302では、ドラッグされたオブジェクトがオブジェクトEであることが認識される（ステップ79__2）。

【0254】

ドラッグされたオブジェクトEがオブジェクトCに重ねられてそこにドロップ

(マウスボタンを離す操作をいう)すると(ステップ79__3)、オブジェクト結合手段302では、そのドロップ先のオブジェクトがオブジェクトCであることが認識される(ステップ79__4)。このようにして、ドラッグされたオブジェクトがオブジェクトEであり、ドロップ先のオブジェクトがオブジェクトCであることが認識されると、オブジェクトツリーが図76に示す状態から図78に示す状態に変更される。

【0255】

この変更は、図76に示すオブジェクトのポインタ群のうち、オブジェクトBの‘TO’に、オブジェクトCへのポインタに代えてオブジェクトEへのポインタが書き込まれ、オブジェクトEの‘FROM’、‘TO’に、それぞれ、オブジェクトBへのポインタ、オブジェクトEへのポインタが書き込まれ、オブジェクトDの‘FROM’に、オブジェクトCへのポインタに代えてオブジェクトEへのポインタが書き込まれることによって行なわれる。

【0256】

次にドロップ先オブジェクトCの親オブジェクトであるオブジェクトAのケーブル要素リストからドロップ先オブジェクトCの配線が検索される(ステップ79__6)。

図80は、オブジェクトAに接続されたケーブル要素リストの一部を示す模式図である。

【0257】

ここには、ケーブル要素CABLE aが指し示すターミナル要素にオブジェクトCのバス3とオブジェクトDのバス4が接続されていることが記録されている。このようにケーブル要素を順次検索して、ドロップ先オブジェクトCに接続された配線が見つけられる。

ドロップ先オブジェクトCに接続された配線が見つけられると、図80に示すように、その配線が解除され、その配線が置き換え後のオブジェクトEの、対応するバスに接続される(ステップ79__7)。尚、置き換え後のオブジェクトEに対応するバスが存在せず配線変更ができなかったときは、表示画面102aにその旨のメッセージが表示され、その配線は消去される。

【0258】

次に、本発明のオブジェクト指向プログラミング支援装置のうちの第3のオブジェクト指向プログラミング支援装置、および本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第3のオブジェクト指向プログラミング用プログラム記憶媒体に格納されたオブジェクト指向プログラミング用プログラムの実施形態について説明する。

【0259】

本発明のオブジェクト指向プログラミング支援装置のうちの第3のオブジェクト指向プログラミング支援装置、および本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第3のオブジェクト指向プログラミング用プログラム記憶媒体に格納されたオブジェクト指向プログラミング用プログラムでは、図64に示す階層構造構築手段303が、操作子304からの、所定の階層に配置されたオブジェクトの中から複数のオブジェクトを指定してそれら複数のオブジェクトを一段下の階層に再配置する旨の指示に応じて、それら複数のオブジェクト相互間の結合構造、およびそれら複数のオブジェクトと、それら複数のオブジェクト以外の他オブジェクトとの間の結合構造が保持されるように、それら複数のオブジェクトが一段下の階層に再配置され、もともとの階層には、それら複数のオブジェクトを内包するオブジェクトが生成されて配置される。

【0260】

図81は、オブジェクト移動前のオブジェクトの結合関係を示す概念図、図82は、図81に示すオブジェクトに関するオブジェクトツリーを示す模式図である。

図82に示すように、配線エディタにオブジェクトAが接続されており、そのオブジェクトAには、その下階層にオブジェクトBが接続され、そのオブジェクトBには、そのオブジェクトBと同階層にオブジェクトCが接続され、そのオブジェクトCには、やはり同階層にオブジェクトDが接続され、そのオブジェクトDには、やはり同階層にオブジェクトEが接続されている。

【0261】

ここで、図81に示すようなオブジェクト間ネットワークを表示画面102a

に表示し、マウス 104 を操作して、一段下の階層へ移動すべきオブジェクトとしてオブジェクト C とオブジェクト D を選択したものとする。

図 8 3 は、オブジェクト移動後のオブジェクトの結合関係を示す概念図、図 8 4 は、図 8 3 に示すオブジェクトに関するオブジェクトツリーを示す模式図である。

【0262】

オブジェクト B が配置された階層と同じ階層にオブジェクト F が生成され、オブジェクト C とオブジェクト E は、オブジェクト F を親オブジェクトとする子オブジェクトとして、オブジェクト F の下階層に配置される。

また、移動前は図 8 1 に示すようにオブジェクト B のバス 3 とオブジェクト C のバス 1 とが直接に接続され、オブジェクト D のバス 3 とオブジェクト E のバス 1 とが直接に接続されていたが、移動後は、図 8 3 に示すように、オブジェクト B のバス 3 はオブジェクト F のバス 1 と接続され、そのオブジェクト F のバス 1 がオブジェクト C のバス 1 と接続され、オブジェクト D とオブジェクト E との接続に関しても、オブジェクト D のバス 3 はオブジェクト F のバス 2 と接続され、オブジェクト F のバス 2 がオブジェクト E のバス 1 と接続されている。

【0263】

図 8 5 は、上記のようなオブジェクトの移動および配線替えの処理を示すフローチャートである。

一段下の階層に移動すべきオブジェクトが選択されると、どのオブジェクト（ここでは図 8 2 に示すオブジェクト C とオブジェクト D）が選択されたかが認識され（ステップ 85__1）、選択されたオブジェクトと同じ階層に、新オブジェクト（この例ではオブジェクト F）が作成され（ステップ 85__2）、選択されたオブジェクト（この例ではオブジェクト C とオブジェクト D）が新オブジェクト（オブジェクト F）の下に置き換えられる（ステップ 85__3）。

【0264】

図 8 6 は、オブジェクトツリーを変更する様子を示す模式図である。

上述のステップ 85__2 においてオブジェクト F が作成されると、オブジェクト B とオブジェクト C との間の接続が断たれてオブジェクト B とオブジェクト F

が同階層に接続され、オブジェクトDとオブジェクトEとの間の接続が断たれてオブジェクトFとオブジェクトEが同階層に接続され、オブジェクトFの下階層にオブジェクトCが接続される。これにより、図84に示すオブジェクト移動後のオブジェクトツリーが完成する。

【0265】

尚、オブジェクトツリーを変更するためのポインタの変更は、図78を参照して説明した場合と同様に行なうことができ、ここでは説明は省略する。

次に、図85のステップ85__4に示すように、選択されたオブジェクト（ここではオブジェクトBとオブジェクトC）の親オブジェクト（ここではオブジェクトA）に接続されたケーブル要素リストから、選択されたオブジェクト（オブジェクトBとオブジェクトC）に接続された配線が検索される（ステップ85__4）。

【0266】

図87は、オブジェクトAに接続されたケーブル要素リストの一部を示す模式図である。

この図87には、ケーブル要素CABLE aに、オブジェクトCのバス4とオブジェクトDのバス1が配線されている旨、およびケーブル要素（CABLE bに、オブジェクトDのバス3とオブジェクトEのバス1とが配線されている旨、示されている。ここで、ケーブル要素CABLE aに示されるオブジェクトCのバス4とオブジェクトDのバス1との配線は、図81に示すように、一段下の階層に移動するよう選択されたオブジェクト（オブジェクトBとオブジェクトC）どうしの間の配線を代表的に示しており、ケーブル要素CABLE bに示されるオブジェクトDのバス3とオブジェクトEのバス1との配線は、一段下の階層に移動されるオブジェクト（オブジェクトD）と、移動されずに元の階層に残るオブジェクト（オブジェクトE）との間の配線を代表的に示したものである。

【0267】

図85のステップ85__4において、上記のようなケーブル要素リストの検索が行なわれると、その検索において見つけられた、選択されたオブジェクト（この例ではオブジェクトBとオブジェクトC）に接続された配線が、新オブジェ

クト（オブジェクト F）の内部のオブジェクト（オブジェクト B とオブジェクト C）どうしの配線であるか、あるいは、新オブジェクト（オブジェクト F）の内部のオブジェクトと外部のオブジェクトとの間の配線であるが判定される（ステップ 8 5 __ 5）。この判定において、新オブジェクト（オブジェクト F）の内部のオブジェクトどうしの配線（図 8 7 に示すケーブル要素 C A B L E a の配線がこれに相当する）であると判定されるとステップ 8 5 __ 6 に進み、その配線が、もともとの親オブジェクト（オブジェクト A）から新オブジェクト（オブジェクト F）に移動される。

【 0 2 6 8 】

図 8 8 は、新オブジェクトへの配線の移動の説明図である。

オブジェクト A に接続されているケーブル要素リストの中からケーブル要素 C A B L E a が取り外され、オブジェクト F に接続されるケーブル要素リストに組み込まれる。

また、図 8 5 のステップ 8 5 __ 5 において、新オブジェクト（オブジェクト F）の内部のオブジェクトと外部のオブジェクトとの配線（図 8 7 に示すケーブル要素 C A B L E b の配線がこれに相当する）であると判定されるとステップ 8 5 __ 7 に進み、新オブジェクト（オブジェクト F）に配線用バスが作成される。

【 0 2 6 9 】

図 8 9 は、オブジェクト F に作成された配線用バスの模式図である。

この図 8 9 では、オブジェクト F の ‘O U T’（図 6 7 参照）にバス要素 B U S 2 が接続されている。このバス要素 B U S 2 は、図 8 3 に示すオブジェクト F のバス 2 に相当し、このバス要素 B U S 2 は、オブジェクト D へのポイントと、オブジェクト D のバス 3 へのポイントを持っている。すなわち、このバス要素 B U S 2 により、図 8 3 に示すように、オブジェクト F のバス 2 とオブジェクト D のバス 3 との間に配線が形成されている。尚、この図 8 9 に示すバス要素 B U S 2 は例示的に示したものであり、図 8 3 に示す配線の場合、オブジェクト F の ‘I N’ にも接続用バス要素が配置されてオブジェクト F のバス 1 とオブジェクト C のバス 1 との間も配線される。

【 0 2 7 0 】

図 8 5 のステップ 8 5 __ 8 では、新オブジェクト（オブジェクト F）の内部のオブジェクトに接続されていた配線が、新オブジェクト（オブジェクト F）につながえられる。

図 9 0 は、新オブジェクト（オブジェクト F）の内部のオブジェクト（オブジェクト D）に接続されていた配線を、オブジェクト F につながかえる様子を示す模式図である。

【 0 2 7 1 】

図 8 7 に示すオブジェクト A のケーブル要素 C A B L E b は、オブジェクト F の内部のオブジェクト D のバス 3 と、オブジェクト F の外部のオブジェクト E のバス 1 との間の配線を示しており、図 9 0 に示すように、オブジェクト D のバス 3 がオブジェクト F のバス 2 に変更され、これにより、オブジェクト F のバス 2 と、オブジェクト E のバス 1 との間の配線が形成される。

【 0 2 7 2 】

尚、図 8 5、ステップ 8 5 __ 4 は、必要な回数繰り返し実行される。

次に、本発明のオブジェクト指向プログラミング支援装置のうちの第 4 のオブジェクト指向プログラミング支援装置、および本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第 4 のオブジェクト指向プログラミング用プログラム記憶媒体に格納されたオブジェクト指向プログラミング用プログラムの実施形態について説明する

本発明のオブジェクト指向プログラミング支援装置のうちの第 4 のオブジェクト指向プログラミング支援装置は、図 6 4 に示す表示手段 3 0 1 が、操作子 3 0 4 による指示に応じて、指定された 1 つのメッセージ端子（メッセージ（インスタクション）を出力するバス）に接続されたメソッド端子（メッセージ（インスタクション）を受け取るバス）が複数存在する場合、これら複数のメソッド端子に対応する複数のメソッドの実行順序をあらわすリストを表示する手段を有し、オブジェクト結合手段 3 0 2 が、操作子 3 0 4 による操作に応じて、表示手段 3 0 1 に表示されたリストにあらわれた複数のメソッドの実行順序が変更された結合構造を再構築する手段を有する。

【 0 2 7 3 】

また、本発明のオブジェクト指向プログラミング用プログラム記憶媒体のうちの第 4 のオブジェクト指向プログラミング用プログラム記憶媒体に格納されたオブジェクト指向プログラミング用プログラムは、オブジェクト結合手段 3 0 2 が、指定された 1 つのメッセージ端子に接続されたメソッド端子が複数存在する場合の、これら複数のメソッド端子に対応する複数のメソッドの実行順序をあらわすリストを作成する手段と、そのリストにあらわれた複数のメソッドの実行順序の変更指示に応じて、それら複数のメソッドの実行順序が変更された結合構造を再構築する手段とを有するものである。

【 0 2 7 4 】

図 9 1 は、オブジェクト間の配線の模式図、図 9 2 は、図 9 1 に示す配線を定義している、ケーブル要素リストを示す模式図である。

図 9 1 に示す例では、オブジェクト A には、オブジェクト B，オブジェクト C，オブジェクト D，およびオブジェクト E が配置されており、オブジェクト B のバス 1 と、オブジェクト C のバス 2、オブジェクト D のバス 2、およびオブジェクト E のバス 1 とが接続されている。ここで、オブジェクト B のバス 1 は、インストラクションを出力するバス（メッセージ端子）であり、オブジェクト C のバス 2、オブジェクト D のバス 2、およびオブジェクト E のバス 1 は、いずれも、インストラクションを受け取るバス（メソッド端子）であるものとする。

【 0 2 7 5 】

これらの間の配線は、図 9 2 に示すように、親オブジェクトであるオブジェクト A に接続されているケーブル要素リストにより定義されている。この図 9 2 に示すケーブル要素リストには、多数のケーブル要素が並んでおり、それらのケーブル要素のうち、ケーブル要素 C A B L E a によりオブジェクト B のバス 1 とオブジェクト C のバス 2 との間の配線が定義され、ケーブル要素 C A B L E b によりオブジェクト B のバス 1 とオブジェクト D のバス 2 との間の配線が定義され、ケーブル要素 C A B L E c によりオブジェクト B のバス 1 とオブジェクト E のバス 1 との間の配線が定義されている。

【 0 2 7 6 】

オブジェクト B から出力されたインストラクション（メッセージ）は 3 つのオ

ブジェクトC, D, Eに伝えられ、各オブジェクトC, D, Eではそれぞれ対応したメソッドが実行されるが、それらのメソッドどうしの実行順序が問題になることがある。例えばオブジェクトBが外部からデータを入力するオブジェクト、オブジェクトCはそのデータに基づいて計算を行なうオブジェクト、オブジェクトDは、その計算結果に基づいてグラフを作成するオブジェクト、オブジェクトEはそのグラフを表示するオブジェクトであった場合、オブジェクトBからの、データの入力終了した旨をあらわすインストラクションにより、先ずオブジェクトC、次にオブジェクトD、最後にオブジェクトEの順序で各メソッドが実行される必要がある。

【0277】

ここで、図91に示す配線図からでは実行順序は不明であり、以下のようにして実行順序が表示され、また、必要に応じてその実行順序が変更される。

図93は、メソッド間の実行順序の表示、およびその実行順序の変更の処理をあらわすフローチャートである。

先ず、例えば図91に示すような画像を表示しておいて、マウス104により所望の配線（ここでは、図91に示す配線）をクリックすること等により、その配線を選択する。そうすると、図64に示すオブジェクト結合手段302では、どの配線が選択されたか認識され（ステップ93__1）、その配線（ケーブル）についてケーブルリストが作成されて表示される（ステップ93__2）。

【0278】

図94は、ケーブルリスト要素リストを示す模式図である。

ケーブルリストの作成にあたっては、図92に示す、親オブジェクト（ここではオブジェクトA）のケーブル要素リストが検索され、その選択された配線を構成しているケーブル要素CABLE a, CABLE b, CABLE cが認識され、ケーブル要素リストに並んだ順に、図94に示すケーブルリスト要素リストを構成するケーブルリスト要素にそのケーブル要素へのポインタが格納される。すなわち、ここに示す例では、図92に示す、その配線を構成する3つのケーブル要素CABLE a, CABLE b, CABLE cそれぞれへのポインタが、この順に、図94に示すケーブルリスト要素リストに並ぶ各ケーブルリスト要素に格

納される。

【0279】

図95は、表示画面102aに表示されるケーブルリストの一例を示す図である。

図94に示すようなケーブルリスト要素リストが作成されると、各2つのオブジェクトの間をつなぐ配線の様子が、そのケーブルリスト要素リストに並んだ順に並べられて表示される。すなわち、図95に示す例では、一番上の行にオブジェクトBのバス1とオブジェクトCのバス2とが接続されている旨が表示され、次の行にオブジェクトBのバス1とオブジェクトDのバス2とが接続されている旨が表示され、さらに次の行にオブジェクトBのバス1とオブジェクトEのバス1とが接続されている旨が表示される。ここでは各行を‘リスト項目’と称する。ここで、このケーブルリストの左側はメッセージ（インストラクション）を発行する側のバス、右側はそのメッセージ（インストラクション）を受け取ってメソッドを実行する側のバスをあらわしている。実際の動作にあたっては、オブジェクトBのバス1からそのバス1に対応したメッセージ（インストラクション）が発行されると、このケーブルリストに示した順序で、各メソッドが実行される。

【0280】

ここで、図93のステップ93__3において、この表示リストに表示された一行分からなるリスト項目がドラッグされたものとする。ここでは、図95に示すケーブルリスト中の一番下の行に示すリスト項目「オブジェクトB：バス1 オブジェクトE：バス1」がドラッグされたものとする。すると、オブジェクト結合手段302（図64参照）では、オブジェクトBのバス1とオブジェクトEのバス1とを接続している配線、すなわち、図92に示すケーブル要素CABLEcによって定義される配線がドラッグされたことが認識される（ステップ93__4）。次にそのドラッグされたリスト項目がドロップされる（ステップ93__5）。ここでは、図95に示すケーブルリスト中の、2番目のリスト項目「オブジェクトB：バス1 オブジェクトD：バス2」の上にドロップされたものとする。すると、オブジェクト結合手段302では、そのドロップ先の配線が、オブジ

ェクトBのバス1とオブジェクトDのバス2とを結ぶ配線であること、すなわち、図92に示すケーブル要素CABLEbによって定義される配線であることが認識される（ステップ93__6）。

【0281】

このようにして、ドラッグされた配線とドロップ先の配線が認識されると、図95に示すケーブルリスト上で、ドラッグされた配線がドロップ先の配線の前に並ぶようにその並びの順序、すなわち実行に際しての実行順序が変更される（ステップ93__7）。

図96は、ケーブル要素リストに並ぶケーブル要素の配列順序が変更される様子を示す模式図、図97は、ケーブル要素の配列順序が変更された後のケーブル要素リストを示す模式図である。

【0282】

各ケーブル要素CABLEは、図69に示すように次のケーブル要素へのポインタを持っており、上記のようにしてリスト項目のドラッグ・ドロップ操作が行なわれると、そのポインタが書き換えられて、ここでの例では、図96に示すように、ケーブル要素CABLEcがケーブル要素CABLEbの前に並ぶようにケーブル要素の配列順序が変更され、ケーブル要素が図97に示すように並んだケーブル要素リストが生成される。

【0283】

図98は、ケーブルリスト要素リストに並ぶケーブルリスト要素の配列順序が変更される様子を示す模式図、図99は、ケーブルリスト要素の配列順序が変更された後のケーブルリスト要素リストを示す模式図である。

上述のようにしてリスト項目のドラッグ・ドロップ操作が行なわれると、図96に示すようにケーブル要素リストに並ぶケーブル要素の配列順序が変更されるが、これに伴って、図98に示すように、ケーブルリスト要素リストに並ぶケーブルリスト要素の配列順序も変更される。ここに示す例では、ケーブル要素CABLEcへのポインタが格納されたケーブルリスト要素が、ケーブル要素CABLEbへのポインタが格納されたケーブルリスト要素の前に並ぶようにケーブルリスト要素の配列順序が変更され、図99に示すケーブルリスト要素リストが生

成される。

【 0 2 8 4 】

図 1 0 0 は、配列順序が変更された後のケーブルリストを示す図である。

上記のような、ケーブル要素およびケーブルリスト要素の配列順序が変更された結果、表示用のケーブルリストも、図 1 0 0 に示すように、リスト項目の順序が変更される。

以上で、図 2 に示すオブジェクト間配線エディタ部 1 2 2 およびその周辺部に関する実施形態の説明を終了し、次に、オブジェクトビルダ部 1 2 1 およびその周辺に関する実施形態の説明に移る。

【 0 2 8 5 】

オブジェクトウェアプログラミングシステムでは、プログラムをオブジェクト化することにより効率の良いプログラミングを行なうことを目指しており、このためには、既存のソフトウェア資産を容易にオブジェクト化することが可能であるか否かが極めて重要である。特に、既存のソフトウェアをそのままオブジェクト化することが可能であれば、利用可能なオブジェクトの数が一挙に飛躍的に増大し、プログラム開発効率は格段に向上する。従来においても、既存のソフトウェアをオブジェクト化する方法が何種類か提案されている。Windows における O L E や D D E などがその例である。しかしながら、これらの方法では、既存ソフトウェア側で、オブジェクト化されることあらかじめ予定してそのように対応していなければ使用できず、すべての既存ソフトウェアをオブジェクト化することができるものではない。また、そのような対応がなされている既存ソフトウェアであっても、受け付けることのできるメッセージの数は、例えばグラフィカル・ユーザインタフェースにおいて受け付けるメッセージの数に比べて大変少ないものが多く、その既存ソフトウェアをグラフィカル・ユーザインタフェースによる場合と同等には操作することはできない。

【 0 2 8 6 】

また、既存ソフトウェア対する連続した操作は、従来より、シェルスクリプトなどによって記述する方法がある。しかしながら、従来手段では、起動後のソフトウェアに対する操作を、グラフィカル・ユーザインタフェースによる場合と同

等に行うのは困難である。また、シェルスクリプトの記述はユーザが自分で行わなければならない、プログラミングの経験のない初心者ユーザには難しい。

【0287】

以下に説明する実施形態は、これらオブジェクト作成上の問題に鑑み、既存ソフトウェアの仕組みによらずに、その既存ソフトウェアオブジェクト化する手法、および、既存ソフトウェアと組み合わせられてオブジェクトとして作用するコンポーネントに関する実施形態である。ここでは、グラフィカル・ユーザインターフェースを持つ既存ソフトウェアを中心として、その既存ソフトウェアをオブジェクト化する手法や、その既存ソフトウェアと組み合わせられてオブジェクトとして作用するコンポーネントについて説明する。

【0288】

尚、以下に説明するコンポーネントと本発明との対応は次の通りである。すなわち以下に説明するコンポーネントが、図1に示すコンピュータシステム100の記憶部105に格納された場合に、そのコンポーネントを記憶した状態における記憶部105が本発明にいうコンポーネント記憶媒体の一例に相当する。また、そのコンポーネントが、図1に示すMO110に格納された場合、そのコンポーネントが格納された状態にあるMO110も、本発明にいうコンポーネント記憶媒体の一例に相当する。

【0289】

図101は、グラフィカル・ユーザインターフェースを持つ既存ソフトウェアを‘包み込む’コンポーネントの一実施形態を示す模式図である。

この図101において、アプリケーションAは表示画面102a（図1参照）上に‘ボタン1’，‘ボタン2’，‘ボタン3’なるアイコンを表示しておき、マウス104の操作によりそれらアイコンのいずれかがクリックされ、そのクリックされたアイコンに応じた処理を実行する既存のソフトウェアである。

【0290】

また、ウィンドウ管理部は、このアプリケーションAを含む、このシステムに組み込まれたアプリケーション全てのグラフィカル・ユーザインターフェースを管理する、例えばWindowsならばそのWindowsシステムそのものを

指す。アプリケーションAを‘包み込む’コンポーネントAは、他オブジェクトと接続されるための、オブジェクトとしての基本構成の他にアプリケーションAに関するデータを持ち、さらに、アプリケーションAを生成（起動）するためのアプリケーション起動プログラムやアプリケーションAに対するウィンドウイベント発生プログラム（例えばユーザがマウス104を操作してボタン1をクリックしたことと同等のことを実行するボタン1クリックイベント発行プログラム等）をメソッドとして持っている。他オブジェクトからコンポーネントAのアプリケーションA起動メソッドにメッセージが送られるとそのメソッドが実行されてアプリケーションAが起動され、そのウィンドウに関する情報（ID情報等）が読み込まれ、コンポーネントAはそのウィンドウ情報を保持する。

【0291】

また、ボタンクリック等のイベントを発行するメソッドに他オブジェクト（もしくは自分自身）からメッセージが送られるとそのメッセージを受け取ったメソッドに記述されているイベント発行プログラムにより、ウィンドウ管理部を経由して、アプリケーションAのウィンドウに対して、対応するイベントが発行される。

【0292】

この場合、既存アプリケーションAには何も手を加えずに、コンポーネントAを追加するだけで、その既存アプリケーションをオブジェクト化することができる。

図102は、グラフィカル・ユーザインターフェースを持つ既存ソフトウェアを‘包み込む’コンポーネントの他の実施形態を示す模式図である。

【0293】

図101を参照して説明した実施形態における、アプリケーションAやその他の同様な既存ソフトウェアに対するイベント発生プログラムの最後に、そのメソッドの実行が終了したことを他オブジェクトに通知するためのメッセージを発行するプログラムを追加する。このメッセージを他コンポーネントないし他オブジェクトのメソッドに接続する。こうすることにより、複数のメソッドを連鎖的に実行することができる。この図102では既存ソフトウェアの図示は省略し、コ

ンポーネントAのメッセージとコンポーネントBおよびコンポーネントCのメソッドとが接続された状態が示されている。

【0294】

図103は、グラフィカル・ユーザインターフェースを持つ既存ソフトウェアを‘包み込む’コンポーネントのもう1つの実施形態を示す模式図である。

この図103に示すコンポーネントは既存ソフトウェアに対するイベントを監視して所定のイベントが発行されたときにそれに対応するメッセージを発行する機能を有するコンポーネントの一例である。

【0295】

既存ソフトウェアであるコンポーネントAの、アプリケーションAを起動するメソッドがメッセージを受け取ると、そのメソッドが実行されてアプリケーションAが起動される。このコンポーネントAは、全てのウィンドウイベントを監視する機能を持ち、発行されたイベントがアプリケーションAに関するイベントであるか否かが調べられ、アプリケーションAに関するイベントが発行された場合に、アプリケーションAに関してそのイベントが発行されたことを他オブジェクト（または自分自身）に通知するためのメッセージを発行する。例えば、ユーザによるマウス104の操作により、表示画面102aに表示されたアプリケーションAに関する‘ボタン1’，‘ボタン2’，‘ボタン3’の3つのアイコンのうち、‘ボタン1’のアイコンがクリックされると、コンポーネントAでは、アプリケーションAの‘ボタン1’のアイコンがクリックされたことが認識され、コンポーネントAはボタン1がクリックされたことを通知するメッセージを発行する。

【0296】

こうすることにより、既存ソフトウェアに直接手を加えて拡張機能を付加することなく、既存ソフトウェアのイベントが発行されたことを受けて、その既存ソフトウェアの仕様にはなかったメソッドを連動して実行させることができる。

図104は、図103に示すウィンドウ管理部内のイベント処理部の構成を示す模式図、図105は、図103に示すコンポーネントAの中のイベント監視部の構成を示す模式図である。

【0297】

ウィンドウ管理部内のイベント処理部は、イベントの発生を受けてそのイベントに対応する処理を行なう部分であり、そのイベント処理部は、各種のイベント処理関数へのポインタがそれぞれ格納された複数のイベント処理要素からなるイベント処理要素リストを持っている。ウィンドウイベントが発生すると、各イベント処理要素に格納されたポインタが指し示す先のイベント処理関数が順次実行される。イベント処理要素リストの最後に並ぶイベント処理要素はデフォルトイベント処理関数を指し示しており、このデフォルトイベント処理関数は、例えばボタンがクリックされたときに表示画面上のボタンをそのボタンが押されたかのように動かす処理等を行なうものである。

【0298】

図103に示すコンポーネントAの、アプリケーションAの起動メソッドの最後に、ウィンドウ管理部に対し、ウィンドウイベントを自分（コンポーネントA）に送るように依頼するプログラムが記述されている。具体的にはウィンドウ管理部の中のイベント処理部が持つイベント処理リスト中に、コンポーネントAのイベント監視部へのポインタが格納されたイベント処理要素を追加する。こうしておくことにより、以後ウィンドウイベントが発生する毎にコンポーネントAのイベント監視部でその発生したイベントを参照することができる。

【0299】

コンポーネントAのイベント監視部には、図105に示すイベントテーブルが格納されている。このイベントテーブルには、アプリケーションAに関する各イベントを定義するウィンドウID、イベントIDや、その他のデータと、そのイベントが発行されたときに発行されるメッセージとの対応関係が記述されている。

【0300】

何らかのウィンドウイベントが発生し、図104に示すウィンドウ管理部を経由してコンポーネントAのイベント監視部にその発生したウィンドウイベントに関するイベントデータが入力されると、そのイベントデータ中のウィンドウIDおよびイベントIDによりイベントテーブルが参照され、そのイベントテーブル

中に、そのイベントデータ中のウィンドウ ID およびイベント ID と一致するウィンドウ ID およびイベント ID が存在するか否か検索が行われる。イベントテーブル中にウィンドウ ID およびイベント ID が一致するものがあったときは、コンポーネント A から、その一致するウィンドウ ID およびイベント ID に対応するメッセージが発行される。

【0301】

図 106 は、本発明のコンポーネント作成装置の基本構成図である。

このコンポーネント作成装置 400 は、第 1 操作子 401、第 2 操作子 402、およびコンポーネント作成手段 403 から構成されている。

第 1 操作子 401 は、メソッドの作成およびメッセージの作成を選択的に指示する操作子である。

また、第 2 操作子 402 は、所定の既存ソフトウェアの、所望のイベントの発行を指示入力する操作子である。

【0302】

尚、ここでは、機能上、第 1 操作子と第 2 操作子とに区別しているが、この区別は機能上のものであって、ハードウェア上は 1 つの操作子であってもよく、図 1 に示すコンピュータシステム 100 では、典型的にはマウス 104 がこれら第 1 操作子と第 2 の操作子との双方に対応する。

また、コンポーネント作成手段 403 は、既存ソフトウェアと合わせて 1 つのオブジェクトとして作用するコンポーネントを作成するものである。具体的には、このコンポーネント作成手段 403 は、第 1 の操作子 401 の操作によりメソッドの作成が指示され、次いで第 2 操作子 402 の操作により既存ソフトウェアの所定のイベントが発行された場合に、そのコンポーネントに、他オブジェクトで発行されたメッセージにより発火してそのイベントを発行するメソッドを作成し、第 1 の操作子 401 によりメッセージの作成が指示され、次いで第 2 操作子 402 により既存ソフトウェアの所定のイベントの発行が指示された場合に、そのコンポーネントに、そのイベントの発生に応答してそのイベントが発生したことを他オブジェクトに通知するためのメッセージを作成するものである。

このコンポーネント作成手段 403 は、図 2 に示すオブジェクトウェアプログ

ラミングシステム102のオブジェクトビルダ部121に相当する。

【0303】

図107は、コンポーネント作成装置の一実施形態を説明するための模式図、図108は、コンポーネント作成装置を用いてコンポーネントを作成する処理を表わすフローチャートである。

オブジェクトビルダ部121は、既存ソフトウェアを‘包み込んで’その既存ソフトウェアと一緒にあって1つのオブジェクトとして作用するコンポーネントを作成するプログラム121aを有し、ユーザからの指示に応じてそのプログラムから‘包み込まれる’既存ソフトウェア（ここではアプリケーションA）を起動し（ステップ108__1）、そのアプリケーションAのウィンドウ情報を獲得してそれを保持する（ステップ108__2）。

【0304】

次に、ユーザにより、その起動したアプリケーションAを‘包み込む’コンポーネントにメソッドを追加するかメッセージを追加するかが選択され、さらに、イベントの種類（例えばボタンクリック、メニュークリック等の区別）が選択される（ステップ108__3）。このメソッド／メッセージの別の選択は、表示画面102a上のメソッドとメッセージのアイコンのいずれか一方をマウスでクリックすることにより行われ、選択した欄に、キーボード103を操作することにより、追加しようとするメソッドもしくはメッセージの名前が登録される。

【0305】

その後、イベントの発生を監視し（ステップ108__4）、マウスを用いたボタンクリック等によりイベントが発生すると（ステップ108__5）、その発生したイベントがアプリケーションAのウィンドウに関するイベントであるか否かが判定され（ステップ108__6）、さらに、ステップ108__3で選択したイベントの種類（ボタンクリック、メニュークリック等の別）と同種のイベントであるか否かが判定される（ステップ108__7）。

【0306】

イベントが発生したときに、その発生したイベントが所望のイベントであるか否かを判定する機構（ウィンドウ管理部とイベント監視部の役割り）は、図10

3～図105を参照して説明した機構と同様であり、ここでは説明は省略する。

今回発生したイベントがアプリケーションAのウィンドウに関するイベントであって、選択したイベントの種類と同種のイベントであったときは、ステップ108__3においてイベントの種類と共に選択されたメソッド／メッセージの別に応じて、そのイベントをメソッドもしくはメッセージとしてコンポーネントAに追加する。すなわち、アプリケーションAを‘包み込む’コンポーネントAに他オブジェクトからメッセージを受け取ったときそのイベントを発生させるメソッドを生成し、あるいは、そのイベントが発生したときに他オブジェクトに向けてそのイベントが発生したことをあらわすメッセージを通知するプログラムを追加する。

【0307】

以上の動作が、ユーザからイベント監視終了指示があるまで継続され（ステップ108__9）、ユーザからのイベント監視終了指示を受けて、起動されていたアプリケーション（ここではアプリケーションA）を終了させ（ステップ108__10）、さらに、このアプリケーションAと、そのアプリケーションAを‘包み込む’コンポーネントAとからなるオブジェクトについて表示および配線用のオブジェクトデータが作成されて、図2に示すオブジェクトデータファイル132に格納され、また、そのオブジェクトがコンパイルされて実行形式のオブジェクトデータが生成され、図2に示す実行型オブジェクトファイルに格納される。このようにして、所望の既存ソフトウェアを‘包み込む’コンポーネントが対話的に作成される。

【0308】

次に、本発明のオブジェクト指向プログラミング支援装置のうちの第5のオブジェクト指向プログラミング装置の一実施形態について説明する。

図109は、図2に示すオブジェクトウェアプログラミングシステム120に、本発明の第5のオブジェクト指向プログラミング支援装置の一実施形態に対応する構成要素が追加されたオブジェクトウェアプログラミングシステムの構成図である。図2に示すオブジェクトウェアプログラミングシステム120の構成要素に対応する構成要素には図2に付した符号と同一の符号を付して示し、相違点

について説明する。

【0309】

この図109に示すオブジェクトウェアプログラミングシステム120'には、図2に示すオブジェクトウェアプログラミングシステム120の構成要素に加え、イベントログ作成部141、コンポーネント結合部142、イベントログファイル151、およびコンポーネントファイル152が備えられている。

図107、図108を参照して説明したコンポーネント作成装置の実施形態においては、作成されたコンポーネントは、オブジェクトデータファイル132および実行型オブジェクトファイル133に格納される旨説明したが、ここでは、実行型オブジェクトファイルに関しては同様であるが、作成されたコンポーネントの表示および配線用のデータは、オブジェクトデータファイル132に代わり、コンポーネントファイル152に格納される。尚、ここでは説明および理解の容易のためにオブジェクトデータファイル132とは別にコンポーネントファイル152を備えた構成としているが、それらのファイルは一体的に構成されていてもよい。

【0310】

ここでは、先ず、図107、図108を参照して説明した手法により、図102を参照して説明した、メッセージを受け取って既存ソフトウェアのイベントを発行し、そのイベントを発行した旨をメッセージとして出力するコンポーネントが、複数の既存ソフトウェアに関し多数作成されてコンポーネントファイル152に格納される。

【0311】

次に、ユーザは、それら多数の既存ソフトウェアを同時もしくは順次に起動させ、種々のイベントを発生させる。すると、イベントログ作成部141により、どのイベントがどの順序で発生したかを表わすイベントログが作成され、イベントログファイル151に書き込まれる。

イベントログの作成が終了すると、コンポーネント結合部142はイベントログファイル151に格納されているイベントログファイルから各イベントを順次読み出してきて、コンポーネントファイル152に格納されたコンポーネントを

、その読み出されたイベントが順次発生するように配線する。

【0312】

この配線結果は、オブジェクト間配線データファイル134に格納され、さらにオブジェクト間配線エディタ部122により必要に応じてさらに配線が行われた後、インタプリタ用配線データに変換されてインタプリタ用配線データファイル135に格納される。

図110は、コンポーネント結合部における処理の流れを示すフローチャート、図111は、その処理の流れを説明するための模式図である。

【0313】

図111に示すように、イベントログファイルには、イベントログ作成部141（図109参照）で作成された、多数のイベントデータが並ぶイベントログが格納されている。また、コンポーネントファイル（図109参照）には、前述したようにしてイベントとメソッドとが対応づけられた多数のコンポーネントが格納されている。

【0314】

コンポーネント結合部では、図110に示すように、イベントログファイルよりイベントを1つロードし（ステップ110__1）、そのロードしたイベントと、コンポーネントファイルに格納されたコンポーネントが持っているイベントとメソッドとの対応関係の記述とが比較され、コンポーネントファイルの中から、そのロードしたイベントと同じイベントが検索される（ステップ110__2）。同じイベントがあったときは、そのイベントに対応するメソッドと、前のメッセージ（以下に説明する）との間が配線される（ステップ110__3）。そして、そのメソッドが実行されることにより発行されるメッセージ（コンポーネントファイルには、図102に示すように、メソッドが実行されるとそのメソッドに対応したイベントが発行されたことを表わすメッセージを発行するタイプのコンポーネントが格納されている。）が「前のメッセージ」として保存される。イベントログファイル141中に格納されたイベントログに次のイベントが残っているときは（ステップ110__4）、ステップ110__1に戻り、次のイベントがロードされ、上記と同様にして配線が行われる。

【0315】

尚、イベントログの一番最初に並んだイベントに関しては「前のメッセージ」は存在しないため、図111に示す、前のメッセージとの間の配線は行わずに、単に、そのイベントを発行するメソッドが実行されたときに発行されるメッセージを「前のメッセージ」として保有することになる。

以上のようにしてコンポーネントどうしを自動配線することができ、このようにして配線しておく、インタプリタ部123における実行の際に、ユーザがイベントログを作成した順序通りに順次に自動的にイベントが発行され、既存ソフトウェアの自動操作が可能となる。

【0316】

尚、ここでは一旦イベントログを作成すると、その作成されたイベントログに並ぶイベントの順序に従って順次に自動配線が行われるが、一旦作成したイベントログを一覧表として表示し、その一覧表の中から必要なイベントをユーザが選択し、そのユーザが選択した順序で自動配線を行なうようにしてもよい。こうすることにより、イベントログを作成する際に間違いがあったときにイベントログを最初から作成し直すことなく、その間違いを訂正することができる。

【0317】

このようにして、アプリケーションの自動操作を実現することにより、例えばWWWブラウザのオートパイロット機能を実現することなどが可能となる。

次に、既存のソフトウェアを‘包み込んで’その既存ソフトウェアをオブジェクト化する手法の他の実施形態、および既存ソフトウェアと組み合わせられてオブジェクトとして作用するコンポーネントの他の実施形態について説明する。

【0318】

図112は、既存ソフトウェアをコンポーネントで‘包み込んだ’状態を示す概念図、図113は、その既存ソフトウェアを‘包み込むための定義表を示す図である。尚、図113中では、オブジェクトを‘LSI’と称している。

ここでは、既存ソフトウェアは、そのみでは単独では実行されず、他のアプリケーションプログラム等から呼び出されて実行される関数ないし関数の集合体からなる既存のプログラムである。

【0319】

図112に示す既存プログラミングには、他のプログラミングから受け取るべきデータ $x_1, x_2, x_3, \dots, x_i, \dots$ が存在し、それら他のプログラムから受け取ったデータを基にある処理を行なう関数 $function1, function2, \dots, functionj, \dots$ が存在し、その処理を行った結果を表わす、他のプログラムに引き渡すべきデータ $y_1, y_2, \dots, y_j, \dots$ が存在する。

【0320】

このような既存プログラムを‘包み込む’にあたり、ここでは図112に示すように、2つのオブジェクトにわけてオブジェクトを定義するものとする。この分け方には、ある程度の任意性があり、ユーザによって分け方が決定される。

ここでは、図112に示すような各種の定義が行われる。まず、(A) ヘッダとして、ここに示す作業ないし処理の全体を特定するためのプロジェクト名やこの処理を実行するための環境が定義される。

【0321】

その(A) ヘッダに、(B) 作成するライブラリの定義、(C) 既存プログラムの定義（どの既存プログラムをオブジェクト化するのかを定義する）、および(D) オブジェクトの定義が続く。(D) オブジェクトの定義は、既存プログラムの中が複数のオブジェクトに分かれるときは、複数存在する。

(D) オブジェクトの定義の中には、他オブジェクトからデータを受け取るポインタを定めるデータ入力用のデータバス（データ入力端子）の定義、実行する処理の入口のポインタを定めるメソッド（メソッド端子）の定義、および処理実行後の、他のオブジェクトに引き渡すべきデータのポインタを定めるデータ出力用のデータバス（データ出力端子）の定義が存在する。尚、ここでは、他のプログラムから呼び出されて処理を実行する既存プログラムを想定しているため、この既存プログラムが他のオブジェクトに対し処理を依頼する（メッセージを発行する）ことは考慮されていない。

【0322】

図113は、図112に示す各種定義を行うための定義項目一覧表を示した図

である。

この一覧表にあらわれたキーワード群は、上述した各種定義を行う一種のプログラム言語である。個々のキーワードの詳細説明は、本発明の本質からは外れるため省略する。

【0 3 2 3】

図 1 1 4 は、定義を行なう際に表示画面 1 0 2 a に表示される画面の一例を示す図である。

この画面の左側には、定義の構造が示されており、かつ、それぞれがアイコンの役割りをなし、それらのアイコンのうちのいずれかをクリックすると、この画面の右側に示すように、そのクリックされたアイコンに対応した項目の定義を行なうための一覧表の枠が表示され、その枠の中を 1 つずつ埋めていくことにより、定義表が完成する。

【0 3 2 4】

このような、一覧表の枠を埋める形式を採用すると対話的に容易に定義を行なうことができる。

定義表が完成すると、既存プログラムと、その既存プログラムに関する定義表からなるコンポーネントは、図 2 に示すオブジェクトビルダ部 1 2 1 により、表示、配線用のデータが抽出されてオブジェクトデータファイル 1 3 2 に格納され、またコンパイラにより実行形式に変換されて実行型オブジェクトファイル 1 3 3 に格納される。

【0 3 2 5】

以上のようにして既存ソフトウェアを、その既存ソフトウェアの形式によらず、かつ、既存ソフトウェア自体に手を加えることなく、このオブジェクトウェアプログラミングシステムにオブジェクトとして取り込むことができる。

【0 3 2 6】

【発明の効果】

以上説明したように、本発明によれば、ソフトウェアの再利用性が格段に高められ、かつ実行速度の速いソフトウェアを実現することができる。

【図面の簡単な説明】

【図 1】

本発明のオブジェクト指向プログラミング装置の一形態を包含するコンピュータシステムの外觀図である。

【図 2】

図 1 に示すコンピュータシステム内に実現されたオブジェクトウェアプログラミングシステムの概要図である。

【図 3】

図 1 に示すコンピュータシステム内に実現されたソフトウェアシステムの第 1 例を示す模式図である。

【図 4】

図 3 に示すオブジェクト A の出力インストラクションバス部とオブジェクト B の入力インストラクションバス部のデータ構造の一例を示す模式図である。

【図 5】

メッセージの発行処理を示すフローチャートである。

【図 6】

図 3 示すオブジェクト結合部の出力インストラクションバス部生成部の処理を示すフローチャートである。

【図 7】

図 3 に示すオブジェクト結合部の入力インストラクションバス部の処理を示すフローチャートである。

【図 8】

図 3 に示すオブジェクト結合部におけるインストラクション結合部の処理を示すフローチャートである。

【図 9】

図 3 に示すオブジェクト A のデータ要素リストのデータ構造の一例を示す模式図である。

【図 1 0】

図 3 に示すオブジェクト結合部のデータ要素生成部の処理を示すフローチャートである。

【図 1 1】

図 3 に示すオブジェクト B のポインタ要素リストのデータ構造の一例を示す模式図である。

【図 1 2】

図 3 に示すオブジェクト結合部のポインタ要素生成部の処理を示すフローチャートである。

【図 1 3】

図 3 に示すオブジェクト結合部のデータ結合部の処理が実行された後の構造を示す模式図である。

【図 1 4】

図 3 に示すオブジェクト結合部のデータ結合部の処理を示すフローチャートである。

【図 1 5】

図 1 に示すコンピュータシステムに実現されたソフトウェアシステムの第 2 例を示す模式図である。

【図 1 6】

図 1 に示すコンピュータシステムに実現されたソフトウェアシステムの第 3 例を示す模式図である。

【図 1 7】

図 1 に示すコンピュータシステムに実現されたソフトウェアシステムの第 4 例を示す模式図である。

【図 1 8】

図 1 に示すコンピュータシステムに実現されたソフトウェアシステムの第 5 例を示す模式図である。

【図 1 9】

図 1 5 ～図 1 8 に示すオブジェクト A の、データ構造の一部を示す模式図である。

【図 2 0】

オブジェクト A のメッセージ発行処理の一例を示すフローチャートである。

【図 2 1】

オブジェクト B の処理中における部分処理の第 1 例を示すフローチャートである。

【図 2 2】

オブジェクト B の処理中における部分処理の第 2 例を示すフローチャートである。

【図 2 3】

オブジェクト B の処理中における部分処理の第 3 例を示すフローチャートである。

【図 2 4】

オブジェクト B の処理中における部分処理の第 4 例を示すフローチャートである。

【図 2 5】

オブジェクト B の処理中における部分処理の第 5 例を示すフローチャートである。

【図 2 6】

オブジェクト B の処理中における部分処理の第 6 例を示すフローチャートである。

【図 2 7】

オブジェクト A のメッセージ発行処理の、図 2 0 に示すメッセージ発行処理例とは異なる例を示すフローチャートである。

【図 2 8】

オブジェクト B の処理中における部分処理の第 7 例を示すフローチャートである。

【図 2 9】

オブジェクト B の処理中の部分処理の第 8 例を示すフローチャートである。

【図 3 0】

オブジェクト B の処理中の部分処理の第 9 例を示すフローチャートである。

【図 3 1】

オブジェクト B の処理中の部分処理の第 10 例を示すフローチャートである。

【図 3 2】

図 1 5 に示すオブジェクト結合部における入力インストラクションタグテーブル生成部の処理を示すフローチャートである。

【図 3 3】

図 1 6 に示すオブジェクト結合部における出力インストラクションタグテーブル生成部の処理を示すフローチャートである。

【図 3 4】

図 1 7 に示すオブジェクト結合部における入力データタグテーブル生成部の処理を示すフローチャートである。

【図 3 5】

図 1 8 に示すオブジェクト結合部における出力データタグテーブル生成部の処理を示すフローチャートである。

【図 3 6】

本発明のオブジェクト間ネットワーク表示方法の一実施形態の表示画面例を示す模式図である。

【図 3 7】

階層化されたネットワークの説明図である。

【図 3 8】

多数のオブジェクトと配線からなる表示画像例を示す図である。

【図 3 9】

サブネットワークを表示した表示画像例を示す図である。

【図 4 0】

サブネットワークの表示方法の他の例を示す図である。

【図 4 1】

複数の升目が連結された表示領域を有する表示画像例を示す図である。

【図 4 2】

配線の表示方法に特徴を持つ表示画像例を示す図である。

【図 4 3】

配線の表示方法の他の例を示す図である。

【図 4 4】

オブジェクトのネットワークを表示するための表示領域を作成する手順の説明図である。

【図 4 5】

ユーザにより、表示画面上にオブジェクトが配置されているときの様子を示す図である。

【図 4 6】

ユーザにより、表示画面上に配置されたオブジェクト間の配線が行われているときの様子を示す図である。

【図 4 7】

サブネットワーク表示前（A）および表示後（B）のオブジェクト間ネットワークの表示画面例を示す図である。

【図 4 8】

図 4 7（A）の表示から図 4 7（B）への表示への変更の手順を示すフローチャートである。

【図 4 9】

サブネットワーク表示手順の説明図である。

【図 5 0】

サブネットワーク表示手順を示すフローチャートである。

【図 5 1】

オブジェクトを示す表示領域を、1つもしくは複数の連結された升目で形成した例を示す模式図である。

【図 5 2】

配線の表示方法の例を示す模式図である。

【図 5 3】

配線の表示例を示す模式図である。

【図 5 4】

図 5 3 に示す配線を実行する手順を示すフローチャートである。

【図 5 5】

配線手順の他の例を示すフローチャートである。

【図 5 6】

配線手順の他の例を示すフローチャートである。

【図 5 7】

配線手順の他の例を示すフローチャートである。

【図 5 8】

図 5 5 ～ 図 5 7 に示す配線手順に従って配線を行なった結果を示す模式図である。

【図 5 9】

図 5 5 ～ 図 5 7 に示す配線手順に従って配線を行なった結果を示す模式図である。

【図 6 0】

図 5 5 ～ 図 5 7 に示す配線手順に従って配線を行なった結果を示す模式図である。

【図 6 1】

図 5 5 ～ 図 5 7 に示す配線手順に従って配線を行なった結果を示す模式図である。

【図 6 2】

図 5 5 ～ 図 5 7 に示す配線手順に従って配線を行なった結果を示す模式図である。

【図 6 3】

図 5 5 ～ 図 5 7 に示す配線手順に従って配線を行なった結果を示す模式図である。

【図 6 4】

本発明のオブジェクト指向プログラミング支援装置およびオブジェクト指向プログラミング用プログラム記憶媒体の基本構成図である。

【図 6 5】

オブジェクトの内包関係の一例を示す概念図である。

【図 6 6】

オブジェクトどうしの、階層構造を規定する接続関係を示す模式図である。

【図 6 7】

1つのオブジェクトの、他のオブジェクトへの接続関係を定めるポインタを示した模式図である。

【図 6 8】

図 6 7 に示す ‘バスへのポインタ’ に接続されるバス要素リストを構成する 1 つのバス要素を示す模式図である。

【図 6 9】

図 6 7 に示す ‘ケーブルへのポインタ’ に接続されるケーブル要素リストを構成するケーブル要素を示す模式図である。

【図 7 0】

オブジェクトの間の配線例を示す模式図である。

【図 7 1】

分身オブジェクトの概念図である。

【図 7 2】

図 7 1 に示すオブジェクトの階層構造（オブジェクトツリー）を示す模式図である。

【図 7 3】

分身オブジェクトの作成処理を示すフローチャートである。

【図 7 4】

本体オブジェクトと分身オブジェクトとの接続関係を示す模式図である。

【図 7 5】

オブジェクトの置き換え前のオブジェクトの結合関係を示す概念図である。

【図 7 6】

図 7 5 に示すオブジェクトに関するオブジェクトツリーを示す模式図である。

【図 7 7】

オブジェクト置き換え後のオブジェクトの結合関係を示す概念図である。

【図 7 8】

オブジェクト置き換え後のオブジェクトツリーの一部を示す模式図である。

【図 7 9】

オブジェクト置き換え処理を示すフローチャートである。

【図 8 0】

オブジェクト A に接続されたケーブル要素リストの一部を示す模式図である。

【図 8 1】

オブジェクト移動前のオブジェクトの結合関係を示す概念図である。

【図 8 2】

図 8 1 に示すオブジェクト群に関するオブジェクトツリーを示す模式図である。

。

【図 8 3】

オブジェクト移動後のオブジェクトの結合関係を示す概念図である。

【図 8 4】

図 8 3 に示すオブジェクト群に関するオブジェクトツリーを示す模式図である。

。

【図 8 5】

オブジェクトの移動および配線替えの処理を示すフローチャートである。

【図 8 6】

オブジェクトツリーを変更する様子を示す模式図である。

【図 8 7】

オブジェクト A に接続されたケーブル要素リストの一部を示す模式図である。

【図 8 8】

新オブジェクトへの配線の移動の説明図である。

【図 8 9】

オブジェクト F に作成された配線用バスの模式図である。

【図 9 0】

新オブジェクト（オブジェクト F）の内部のオブジェクト（オブジェクト D）に接続された配線をオブジェクト F につなぎかえる様子を示す模式図である。

【図 9 1】

オブジェクト間の配線の模式図である。

【図 9 2】

図 9 1 に示す配線を定義している、ケーブル要素リストを示す模式図である。

【図 9 3】

メソッド間の実行順序の表示、およびその実行順序の変更の処理をあらわすフローチャートである。

【図 9 4】

ケーブルリスト要素リストを示す模式図である。

【図 9 5】

表示画面 1 0 2 a に表示されるケーブルリストの一例を示す図である。

【図 9 6】

ケーブル要素リストに並ぶケーブル要素の配列順序が変更される様子を示す模式図である。

【図 9 7】

ケーブル要素の配列順序が変更された後のケーブル要素リストを示す模式図である。

【図 9 8】

ケーブルリスト要素リストに並ぶケーブルリスト要素の配列順序が変更される様子を示す模式図である。

【図 9 9】

ケーブルリスト要素の配列順序が変更された後のケーブルリスト要素リストを示す模式図である。

【図 1 0 0】

配列順序が変更された後のケーブルリストを示す図である。

【図 1 0 1】

グラフィカル・ユーザインターフェースを持つ既存ソフトウェアを' 包み込む' コンポーネントの一実施形態を示す模式図である。

【図 1 0 2】

グラフィカル・ユーザインターフェースを持つ既存ソフトウェアを' 包み込む

’ コンポーネントの他の実施形態を示す模式図である。

【図 1 0 3】

グラフィカル・ユーザインターフェースを持つ既存ソフトウェアを’ 包み込む

’ コンポーネントのもう 1 つの実施形態を示す模式図である。

【図 1 0 4】

図 1 0 3 に示すウィンドウ管理部内のイベント処理部の構成を示す模式図である。

【図 1 0 5】

図 1 0 3 に示すコンポーネント A 中のイベント監視部の構成を示す模式図である。

【図 1 0 6】

本発明のコンポーネント作成装置の基本構成図である。

【図 1 0 7】

コンポーネント作成装置の一実施形態を説明するための模式図である。

【図 1 0 8】

コンポーネント作成装置を用いてコンポーネントを作成する処理を表わすフローチャートである。

【図 1 0 9】

図 2 に示すオブジェクトウェアプログラミングシステムに、本発明の第 5 のオブジェクト指向プログラミング支援装置の一実施形態に対応する構成要素が追加されたオブジェクトウェアプログラミングシステムの構成図である。

【図 1 1 0】

コンポーネント結合部における処理の流れを示すフローチャートである。

【図 1 1 1】

コンポーネント結合部における処理の流れを説明するための模式図である。

【図 1 1 2】

既存ソフトウェアをコンポーネントで’ 包み込んだ’ 状態を示す概念図である。

【図 1 1 3】

図 1 1 2 に示す各種定義を行うための定義項目一覧表を示した図である。

【図 1 1 4】

定義を行なう際に表示画面 1 0 2 a に表示される画面の一例を示す図である。

【符号の説明】

- 1 0 オブジェクト結合部
- 1 1 出力インストラクションバス部生成部
- 1 2 入力インストラクションバス部生成部
- 1 3 インストラクション結合部
- 1 3 a 経路
- 1 4 データ要素リスト生成部
- 1 5 ポインタ要素リスト生成部
- 1 6 データ結合部
- 1 6 a 経路
- 2 0 オブジェクト結合部
- 2 1 出力インストラクションバス部生成部
- 2 2 入力インストラクションバス部生成部
- 2 3 インストラクション結合部
- 2 3 a 経路
- 2 4 入力インストラクションタグテーブル生成部
- 3 0 オブジェクト結合部
- 3 1 出力インストラクションバス部生成部
- 3 2 入力インストラクションバス部生成部
- 3 3 インストラクション結合部
- 3 3 a 経路
- 3 4 出力インストラクションタグテーブル生成部
- 4 0 オブジェクト結合部
- 4 1 出力インストラクションバス部生成部
- 4 2 入力インストラクションバス部生成部

- 4 3 インストラクション結合部
 - 4 3 a 経路
- 4 4 入力データタグテーブル生成部
- 5 0 オブジェクト結合部
- 5 1 出力インストラクションバス部生成部
- 5 2 入力インストラクションバス部生成部
- 5 3 インストラクション結合部
 - 5 3 a 経路
- 5 4 出力インストラクションタグテーブル生成部
- 1 0 0 コンピュータシステム
 - 1 0 1 本体部
 - 1 0 2 画像表示部
 - 1 0 2 a 表示画面
 - 1 0 3 キーボード
 - 1 0 4 マウス
 - 1 0 5 記憶部
 - 1 1 0 MO
 - 1 2 0 オブジェクトウェアプログラミングシステム
 - 1 2 1 オブジェクトビルダ部
 - 1 2 2 オブジェクト間配線エディタ部
 - 1 2 3 インタプリタ部
 - 1 3 1 既存ソフトウェアファイル
 - 1 3 2 オブジェクトデータファイル
 - 1 3 3 実行型オブジェクトファイル
 - 1 3 4 オブジェクト間配線データファイル
 - 1 3 5 インタプリタ用配線データファイル
 - 1 4 1 イベントログ作成部
 - 1 4 2 コンポーネント結合部
 - 1 5 1 イベントログファイル

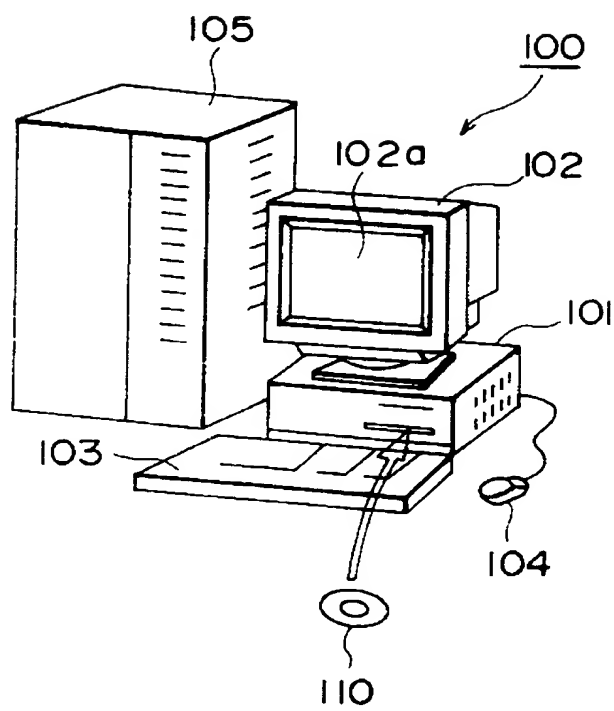
- 1 5 2 コンポーネントファイル
- 2 0 1 格子
- 2 0 2 表示画面
- 2 0 3 オブジェクト表示部分領域
- 2 0 4 配線表示部分領域
- 2 0 5 オブジェクト
- 2 0 6 サブネットワーク
- 2 0 7 表示画像
- 2 0 8 出力端子
- 2 0 1, 2 1 1, 2 1 2, 2 1 3, 2 1 4, 2 1 5, 2 1 6, 2 1 7, 2 1 8
- , 2 1 9 オブジェクト
- 2 2 0 出力端子
- 2 2 1 入力端子
- 2 2 2 配線
- 2 2 3 配線
- 2 2 3 a 中心線
- 2 2 3 b 緑線
- 2 2 4 オブジェクト
- 2 2 5 領域
- 2 2 6, 2 2 7 オブジェクト
- 2 2 8 操作
- 2 2 9 配線
- 3 0 0 オブジェクト指向プログラミング支援装置
- 3 0 1 表示手段
- 3 0 2 オブジェクト結合手段
- 3 0 3 階層構造構築手段
- 3 0 4 操作子
- 4 0 0 コンポーネント作成装置
- 4 0 1 第 1 操作子

4 0 2 第 2 操作子

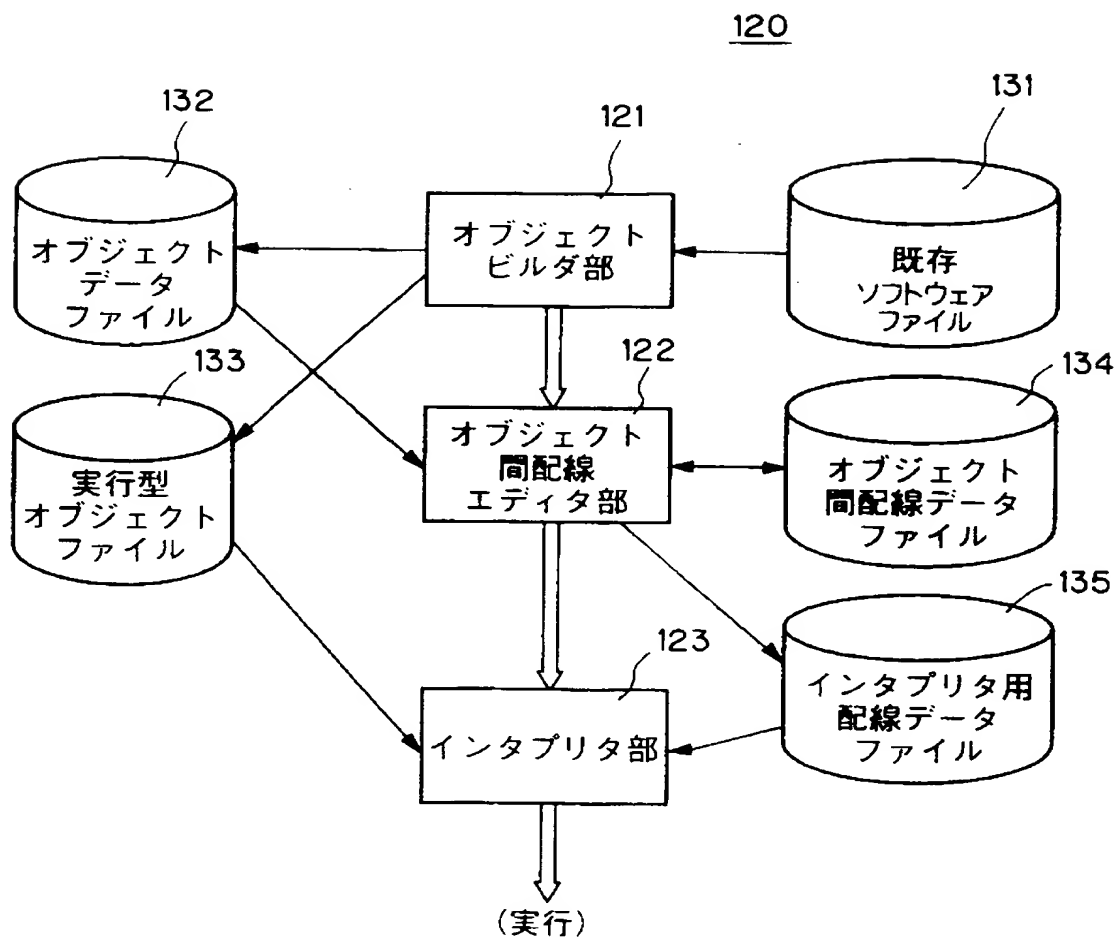
4 0 3 コンポーネント作成手段

【書類名】 図面

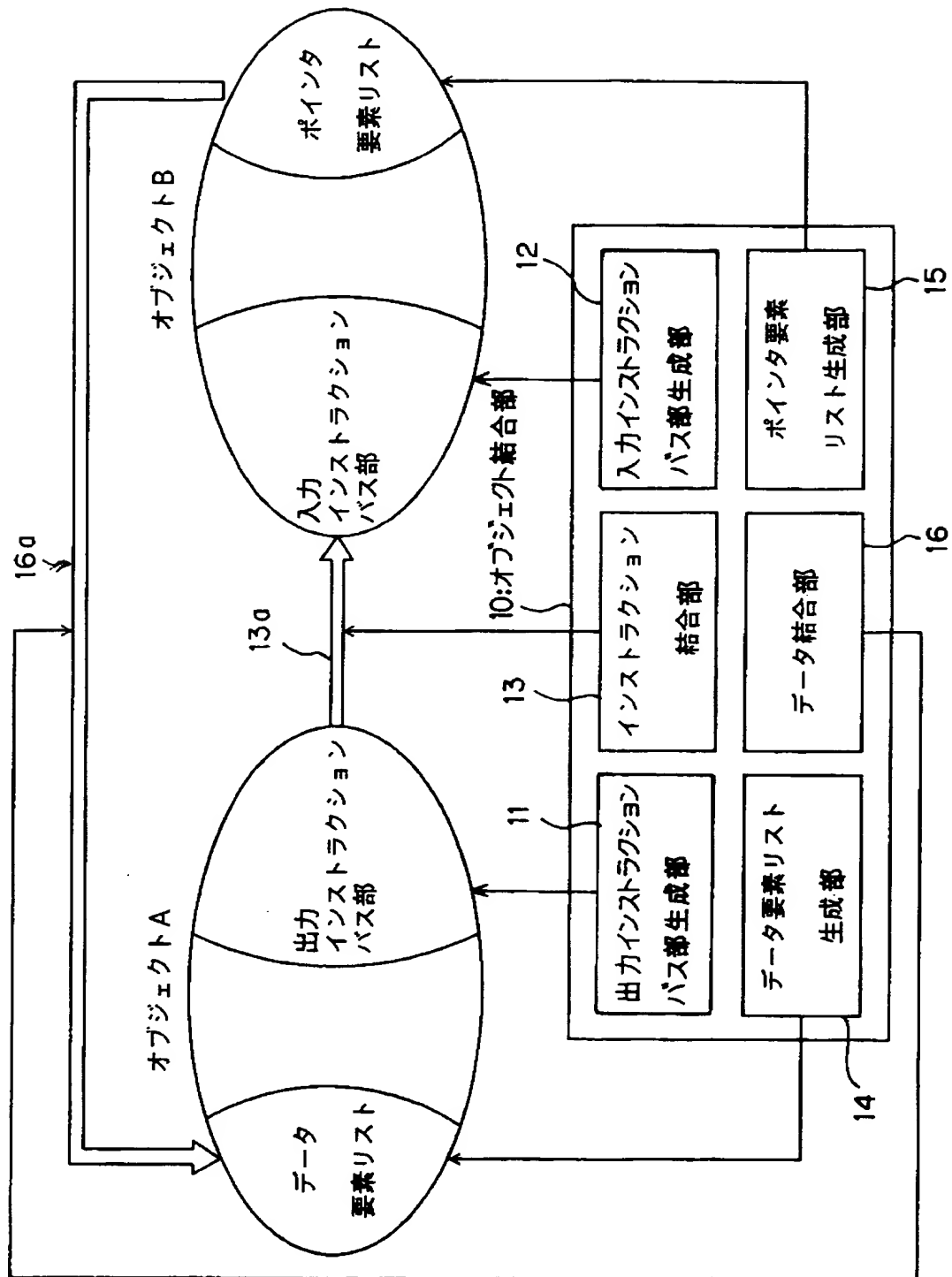
【図 1】



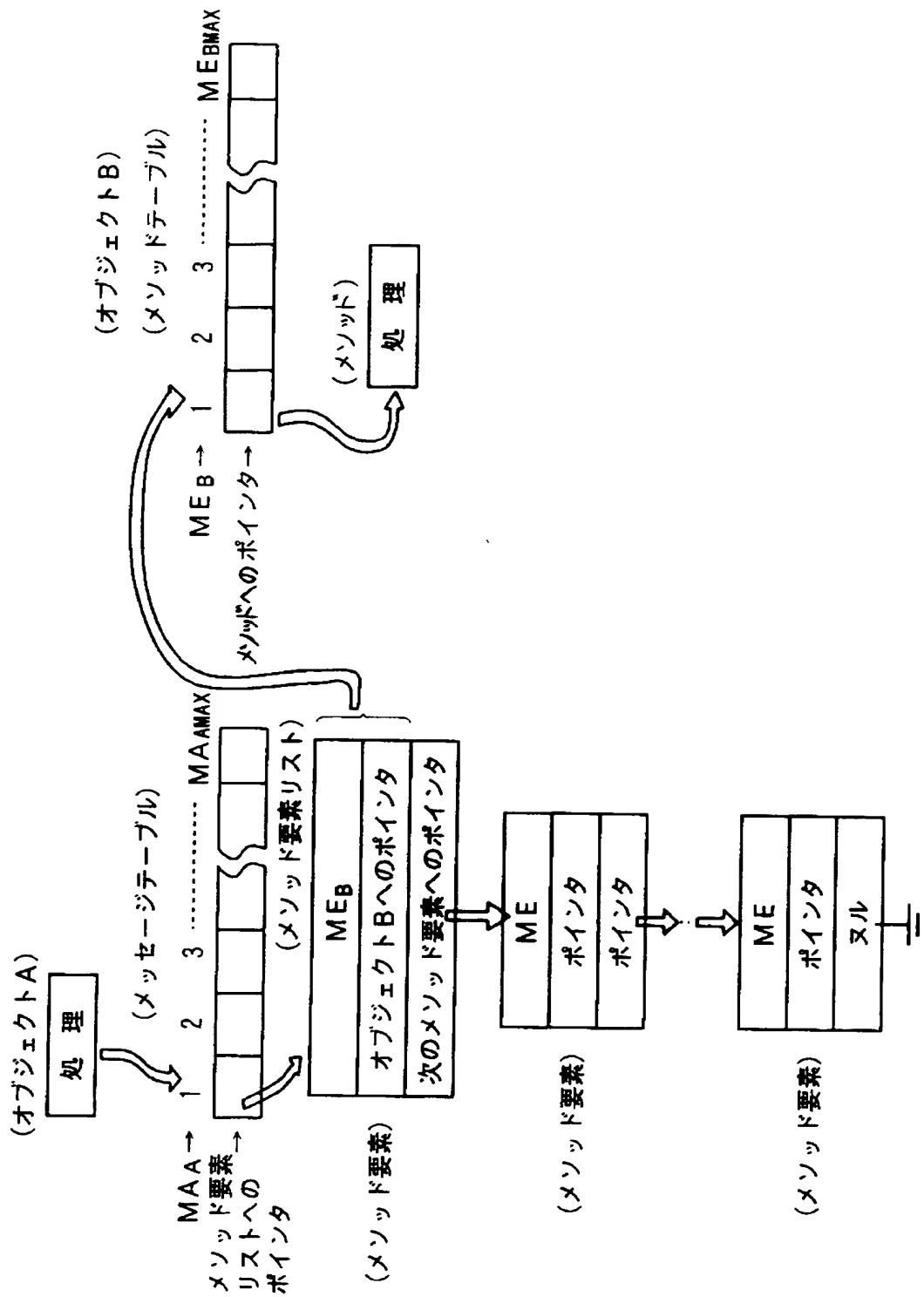
【図2】



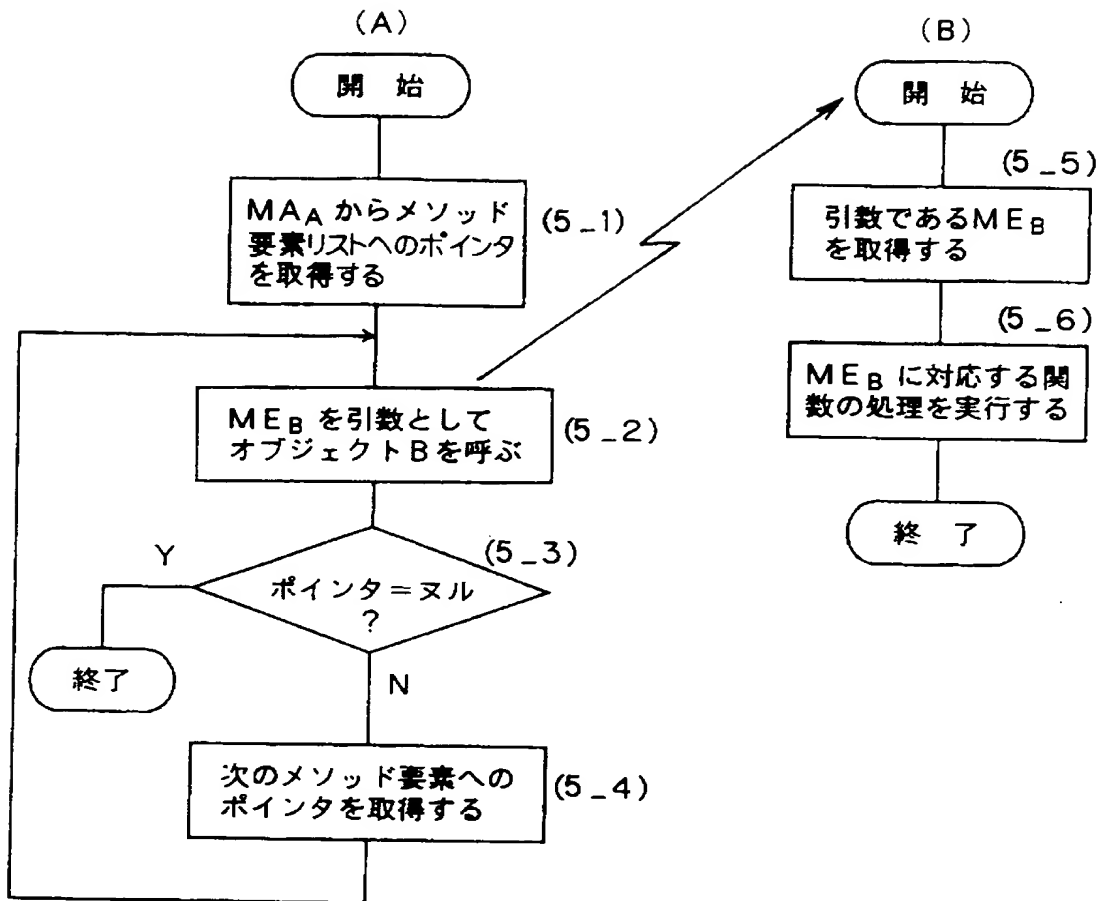
【図3】



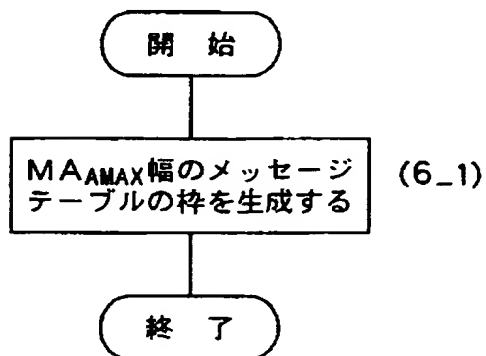
【図4】



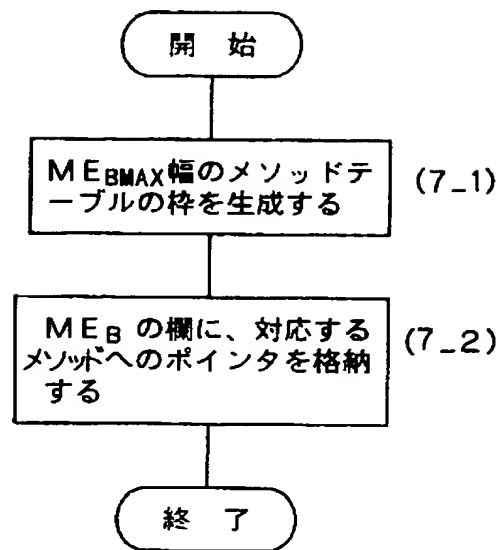
【図5】



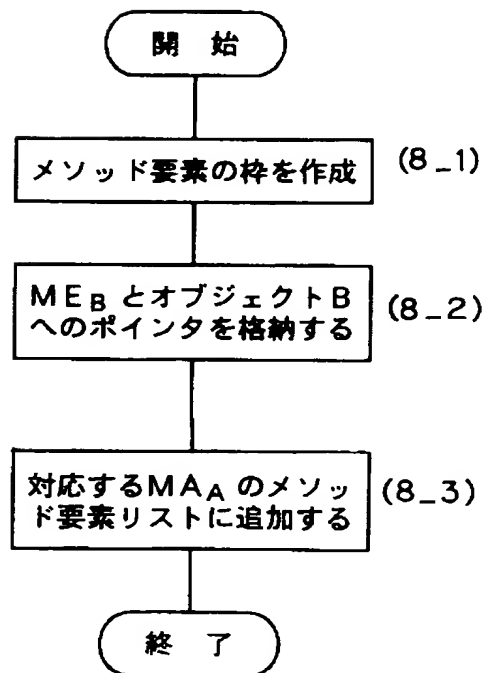
【図6】



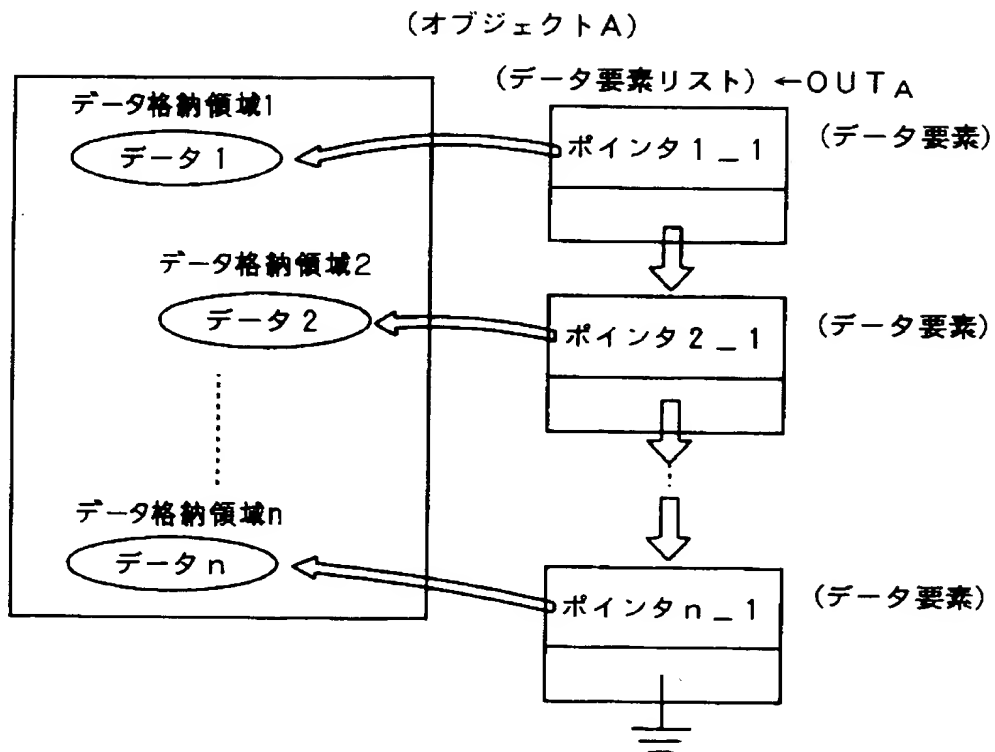
【図7】



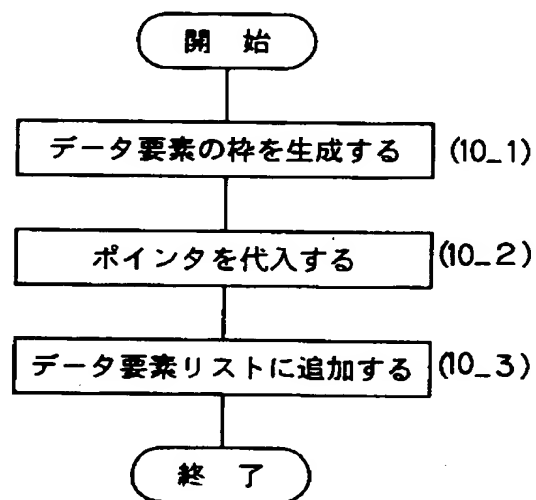
【図8】



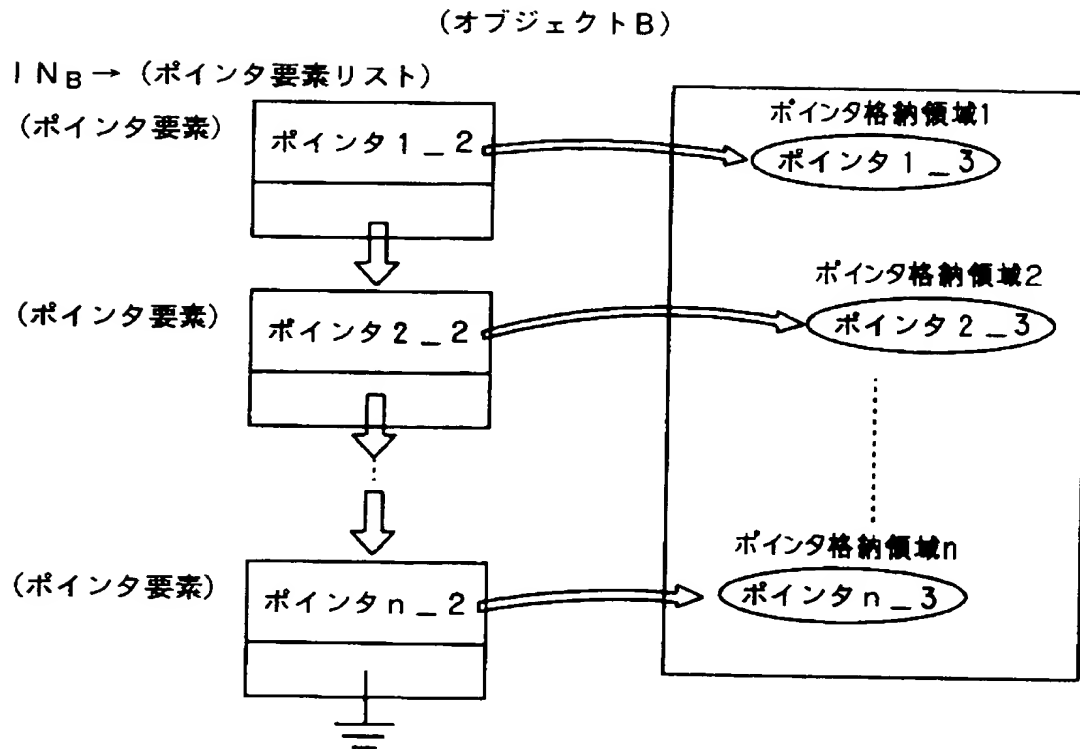
【図9】



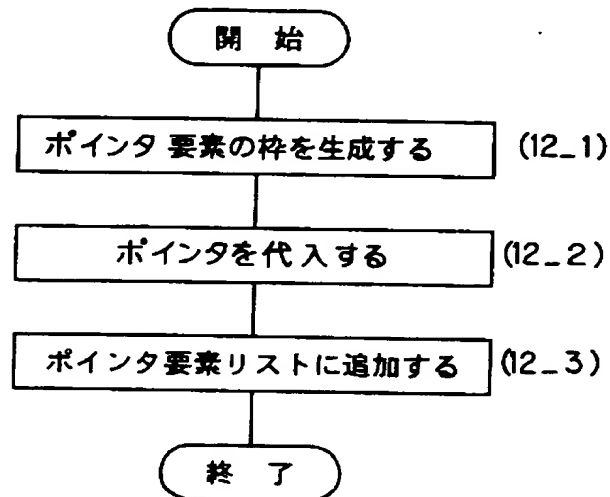
【図10】



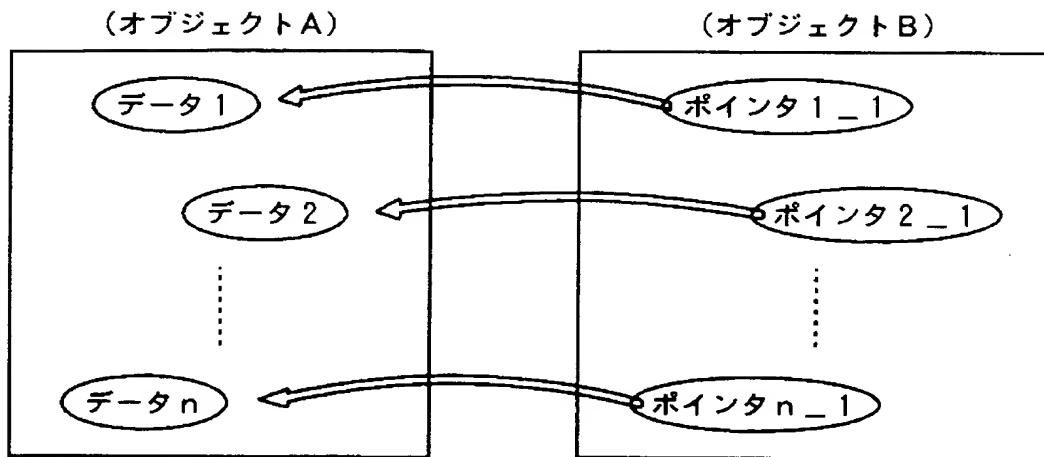
【図11】



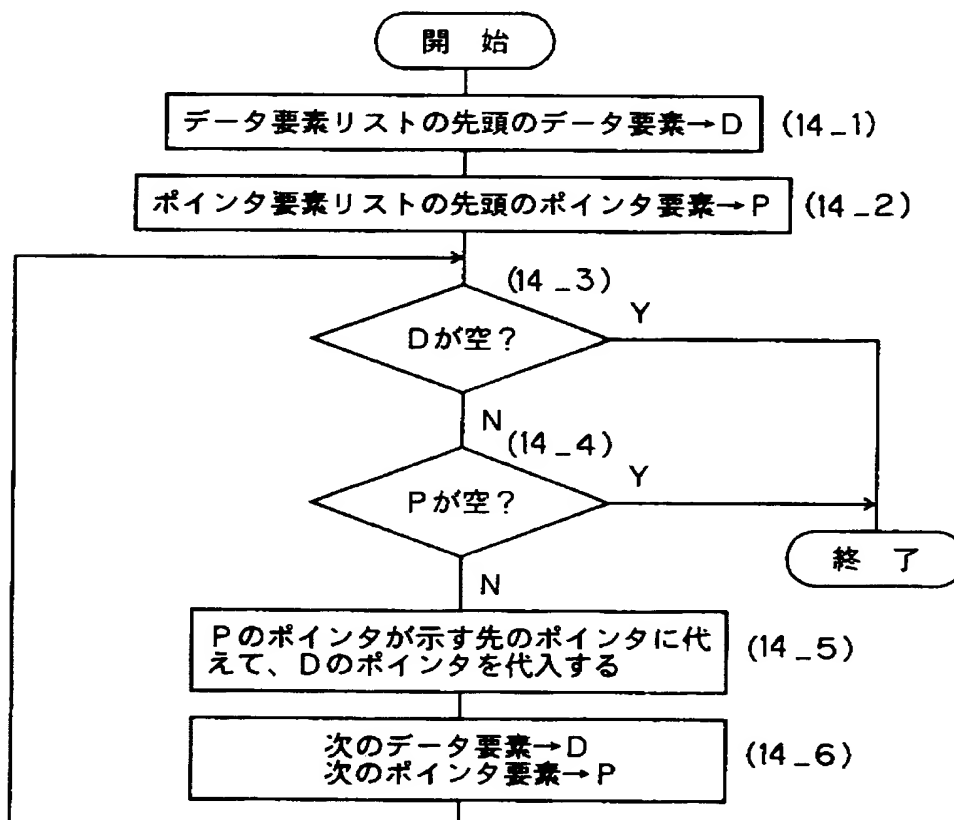
【図12】



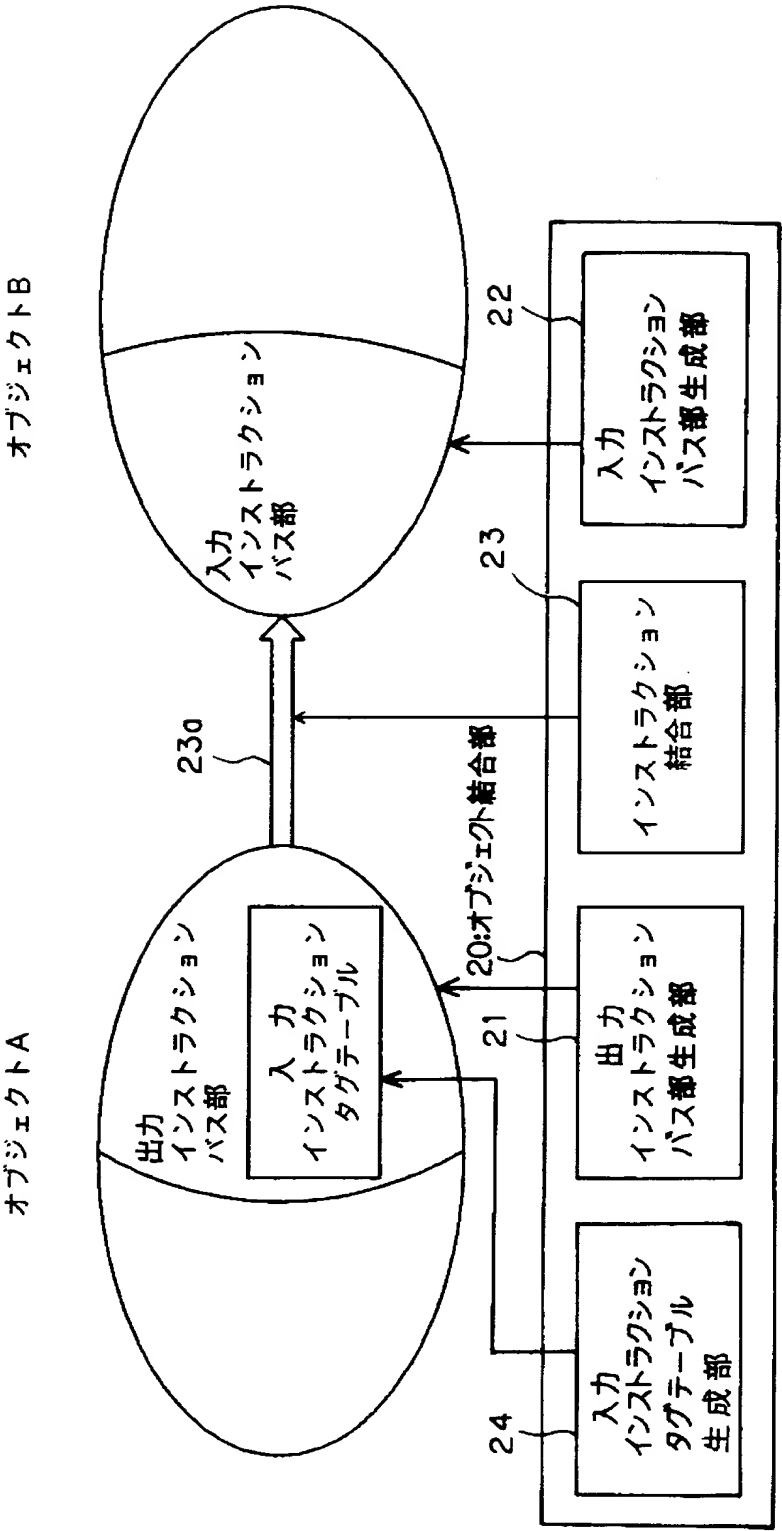
【図13】



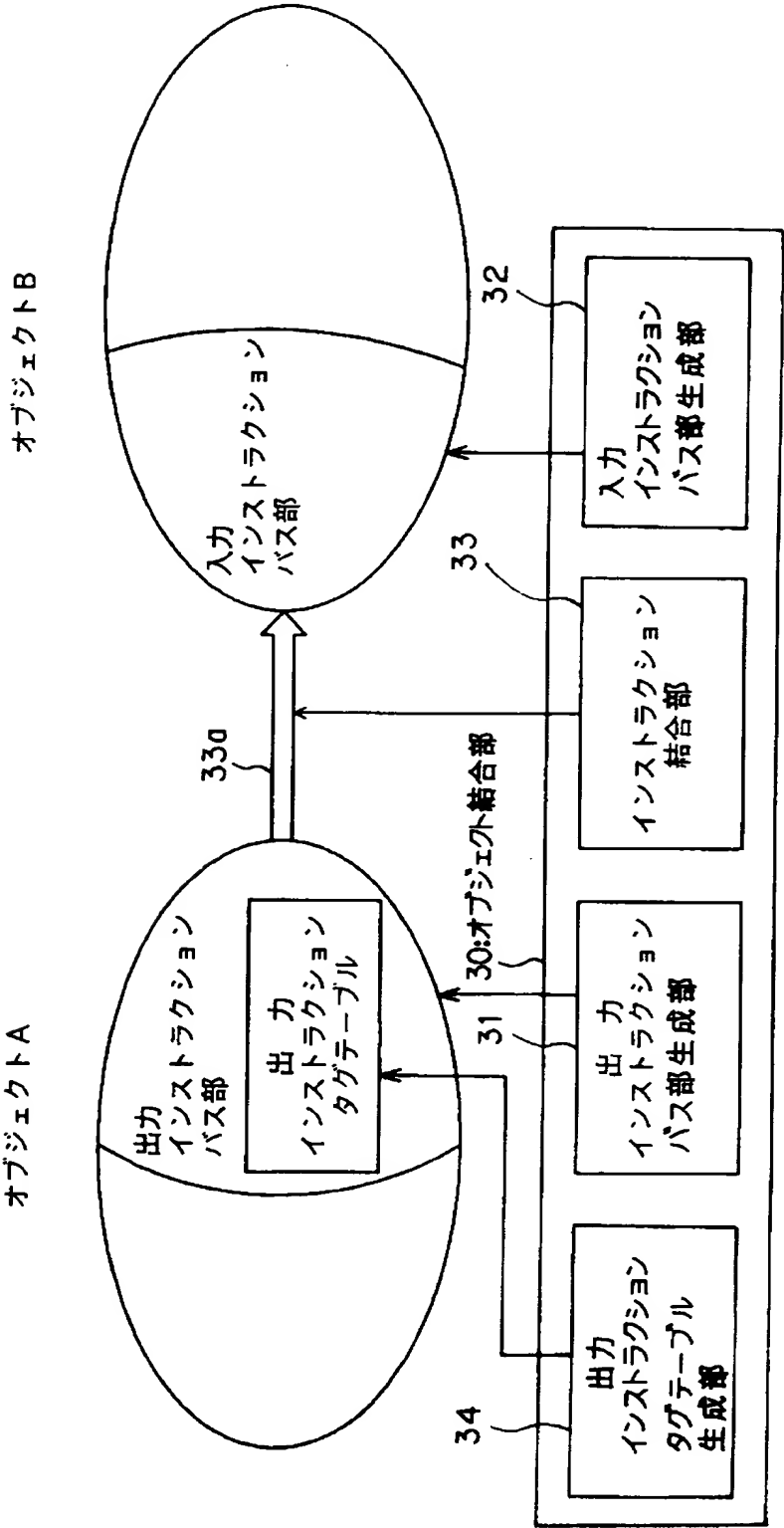
【図14】



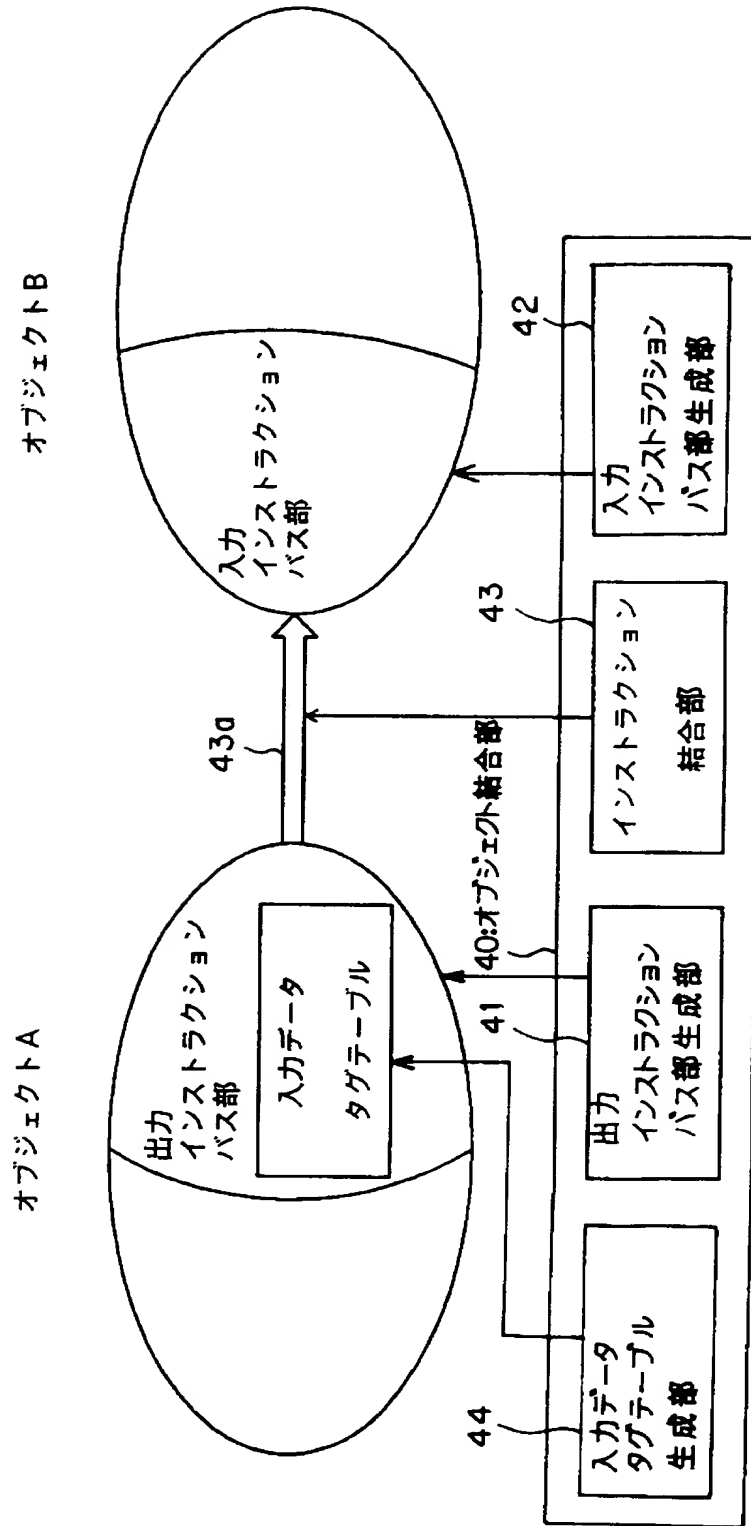
【図15】



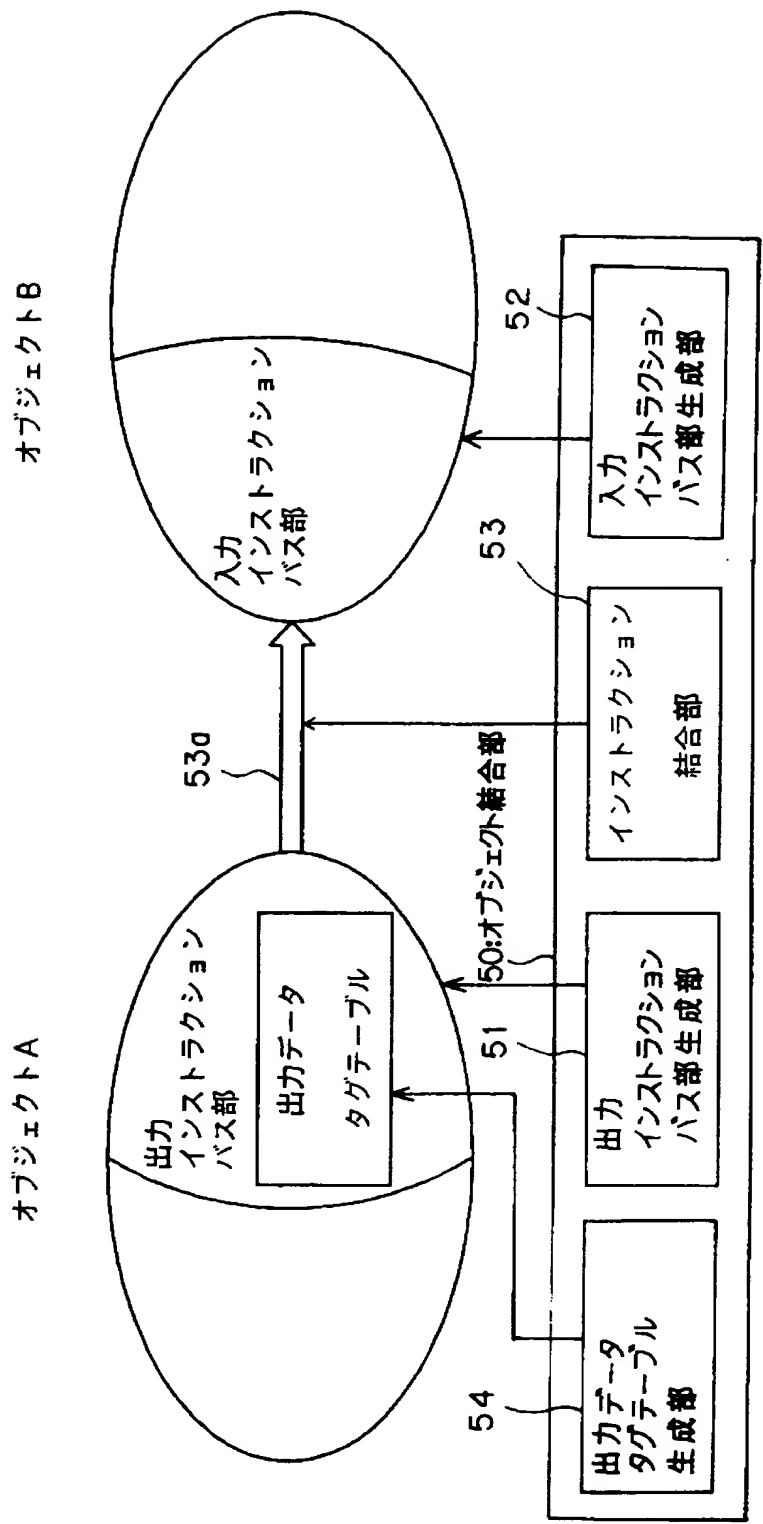
【図16】



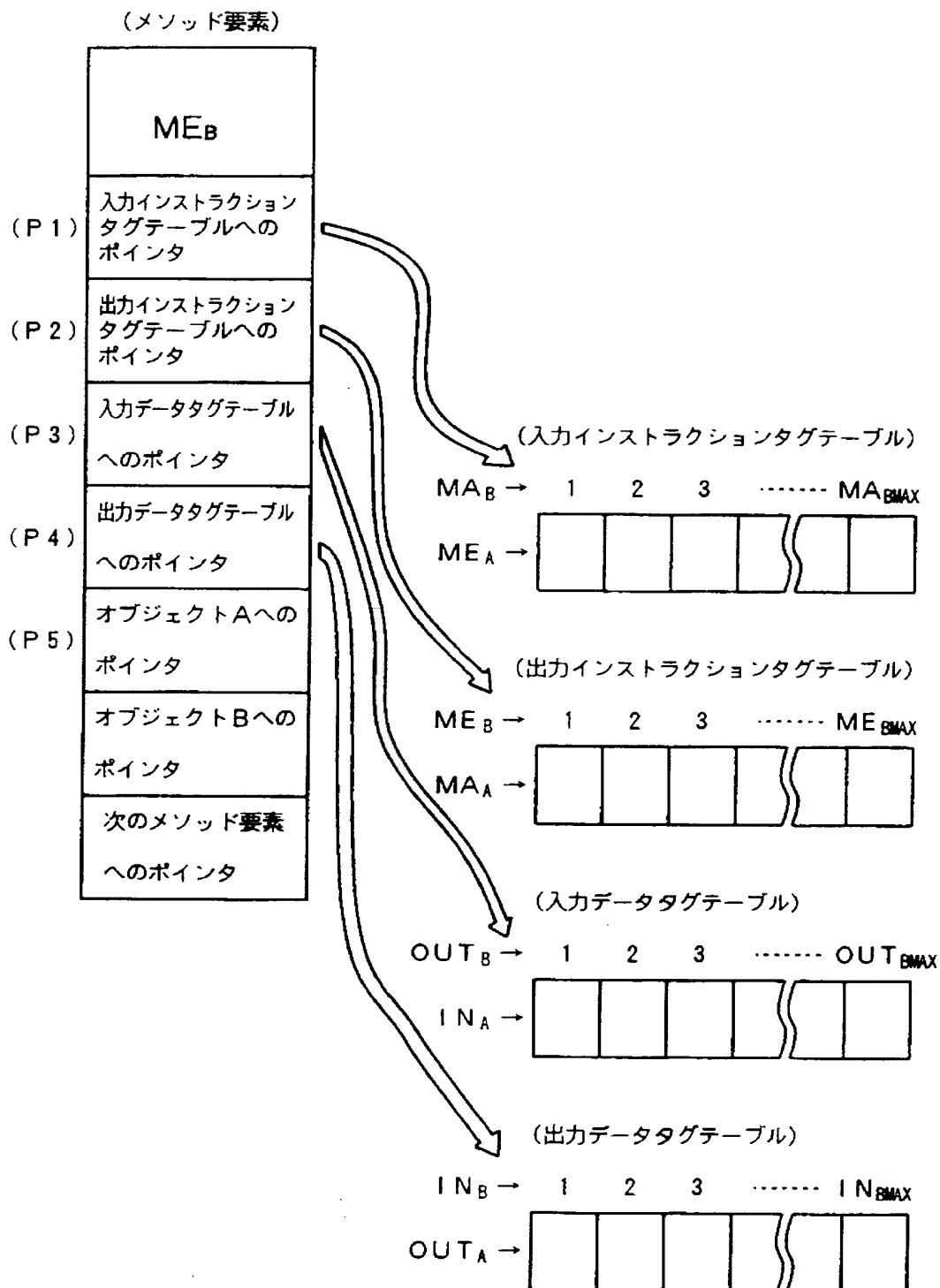
【図17】



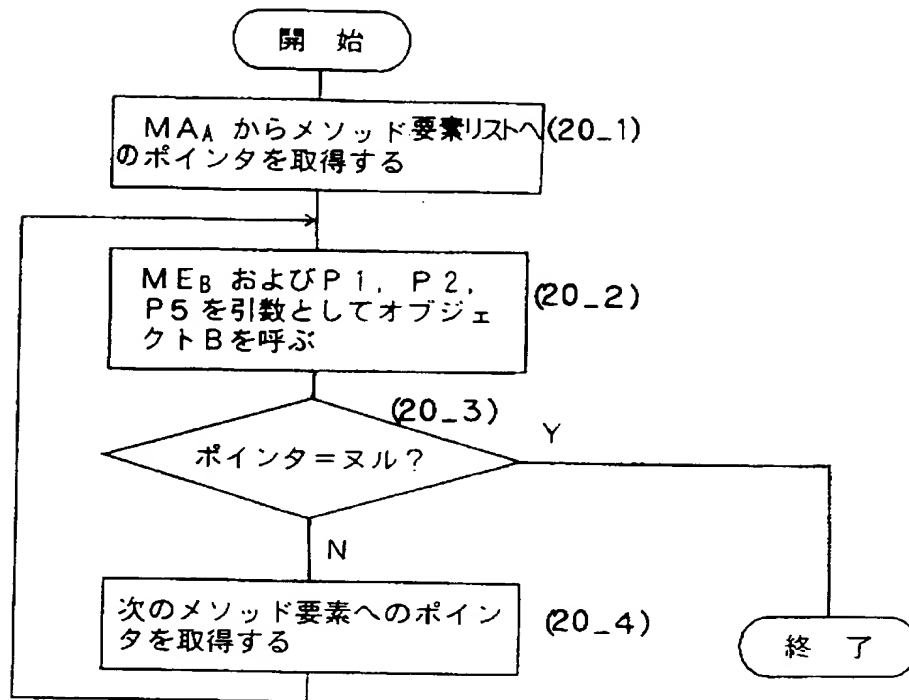
【図18】



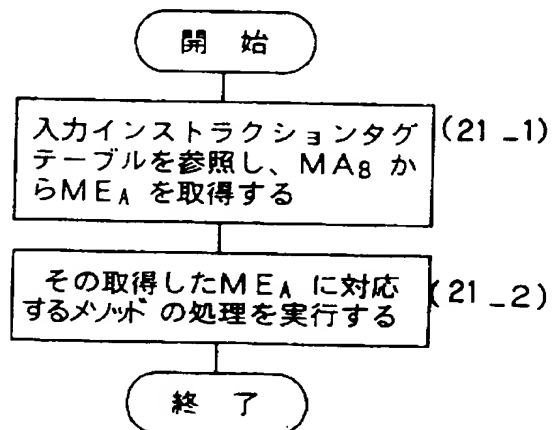
【図19】



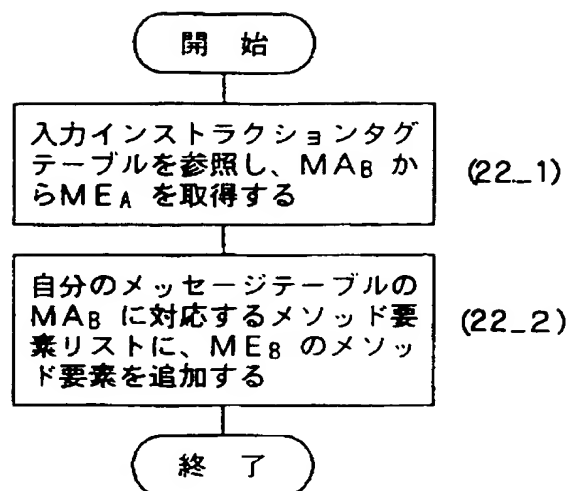
【図20】



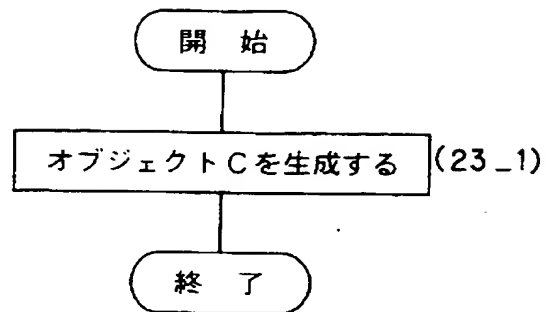
【図21】



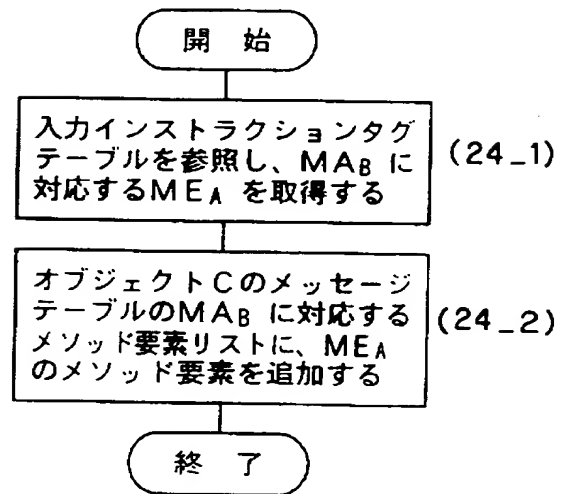
【図 22】



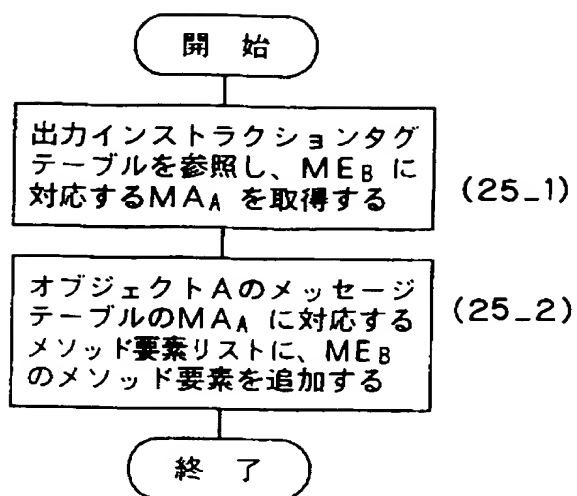
【図23】



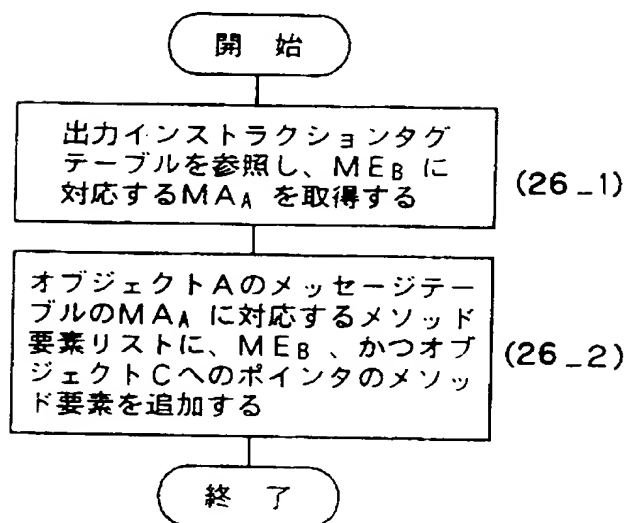
【図24】



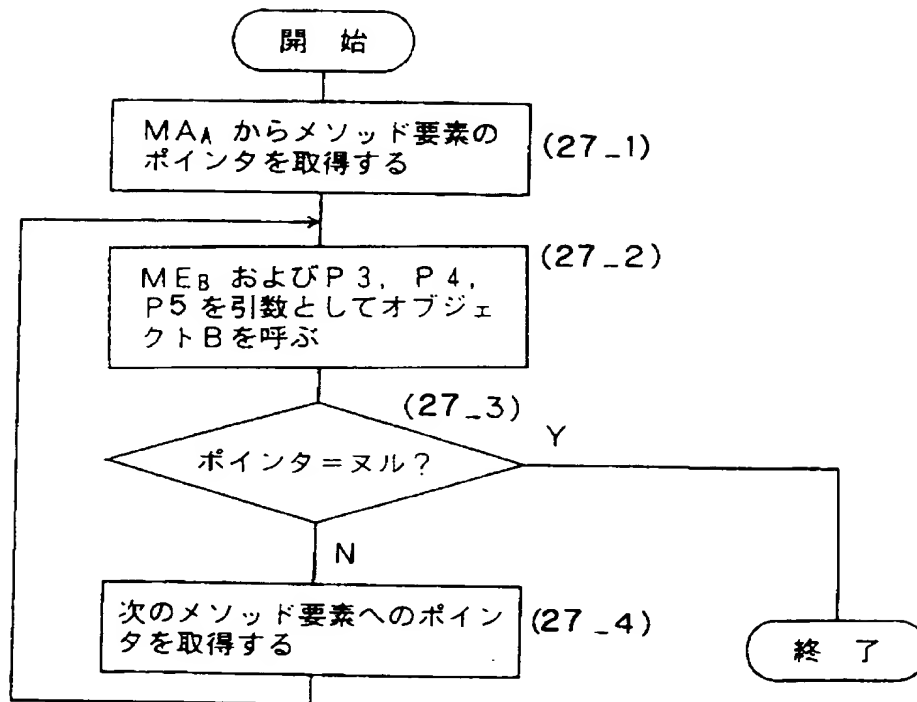
【図25】



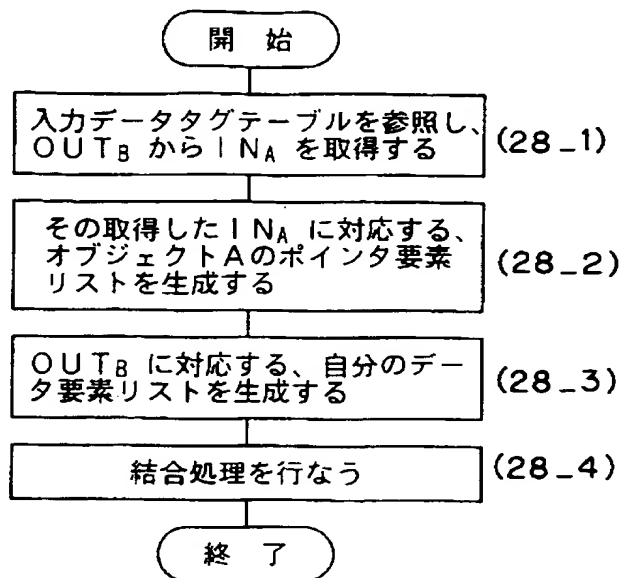
【図26】



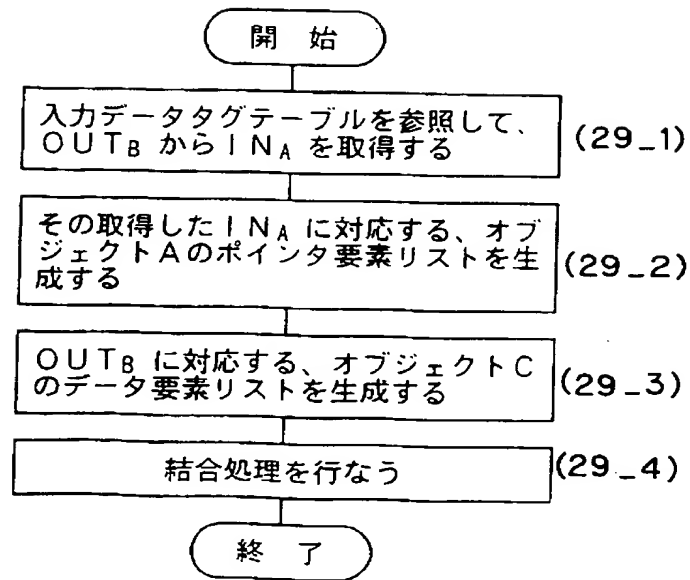
【図27】



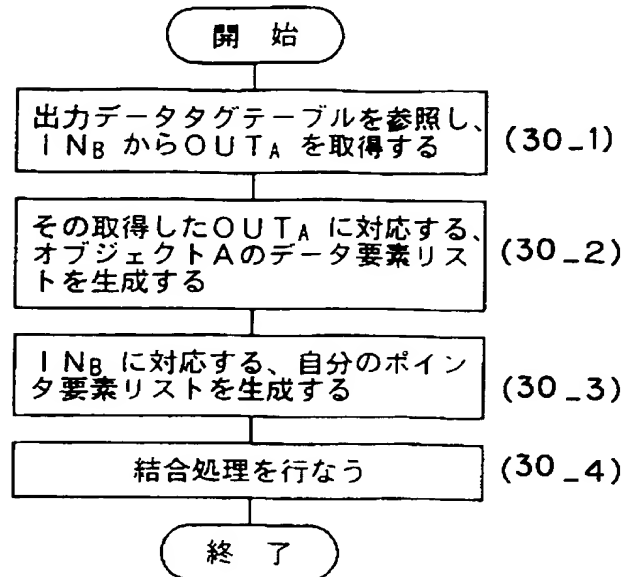
【図28】



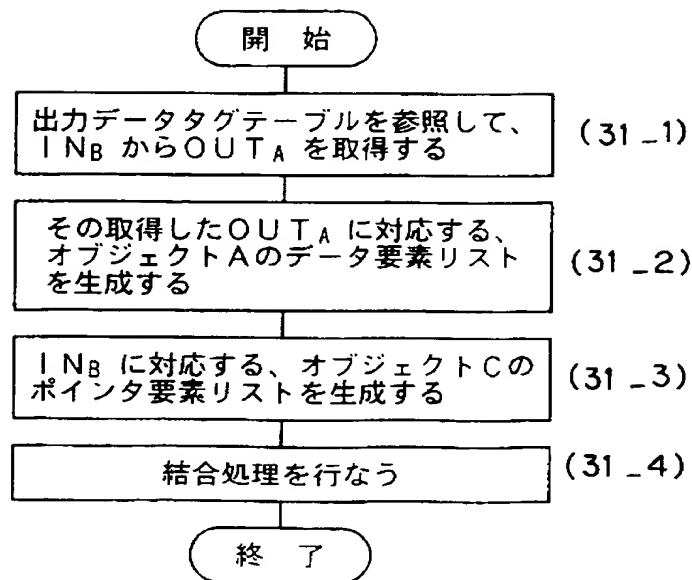
【図29】



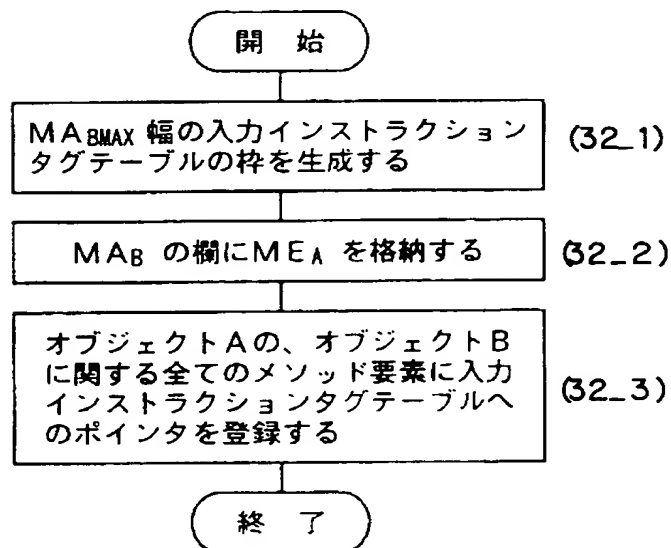
【図30】



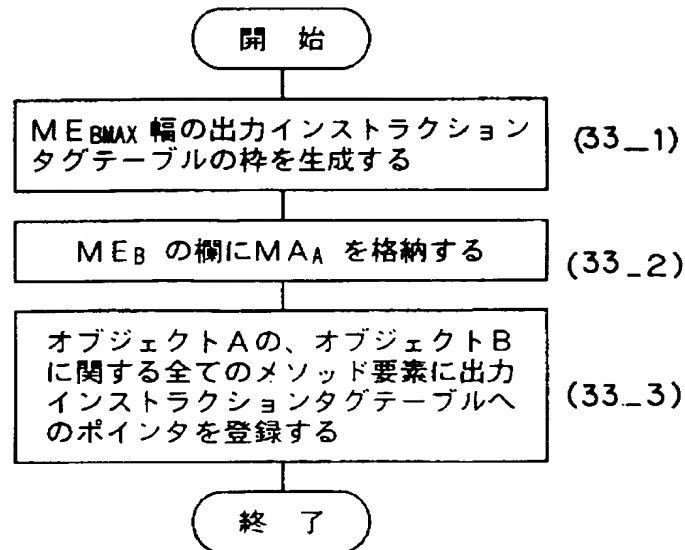
【図31】



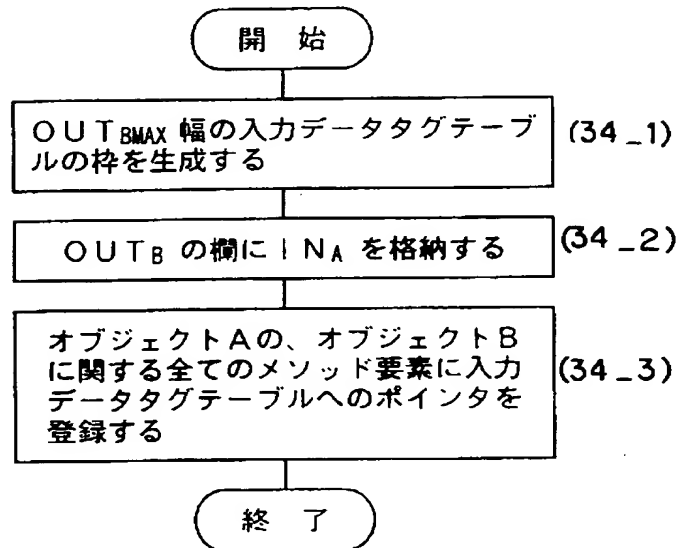
【図32】



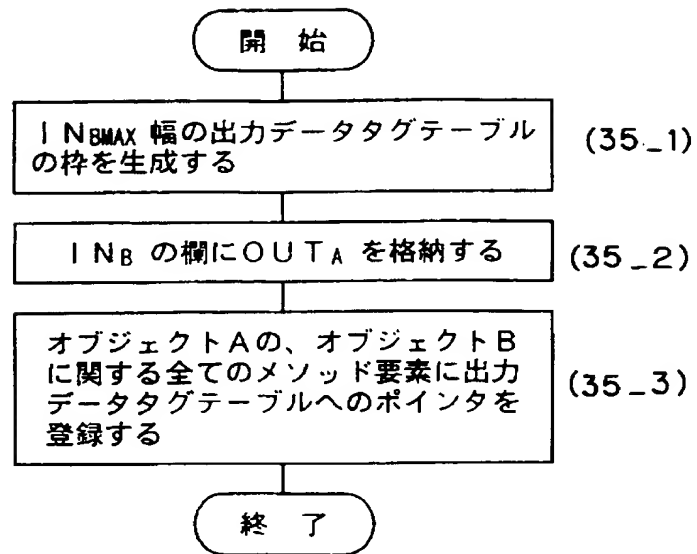
【図33】



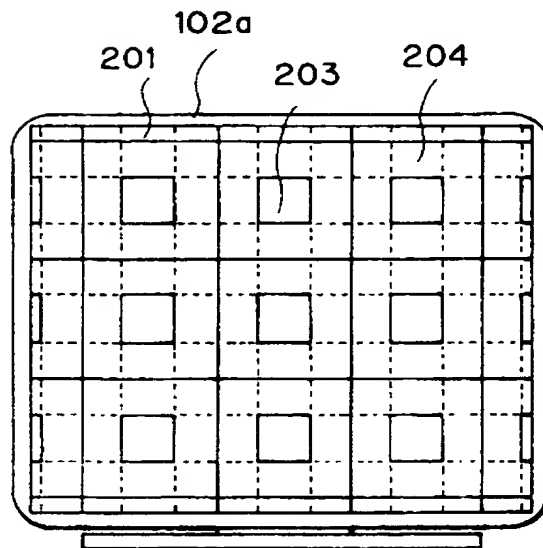
【図34】



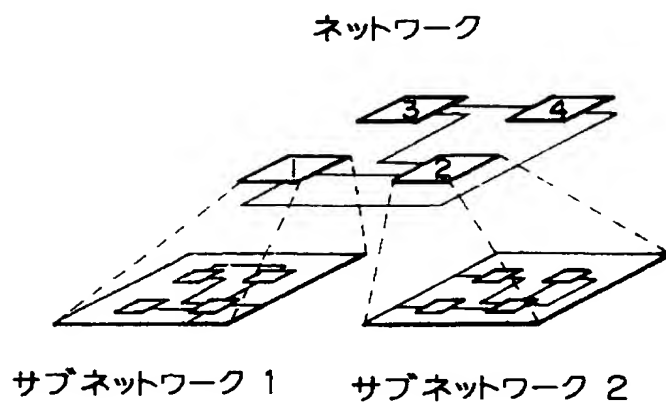
【図 3 5】



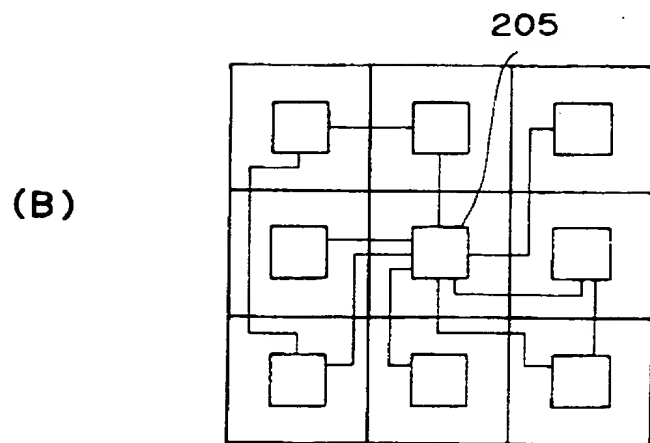
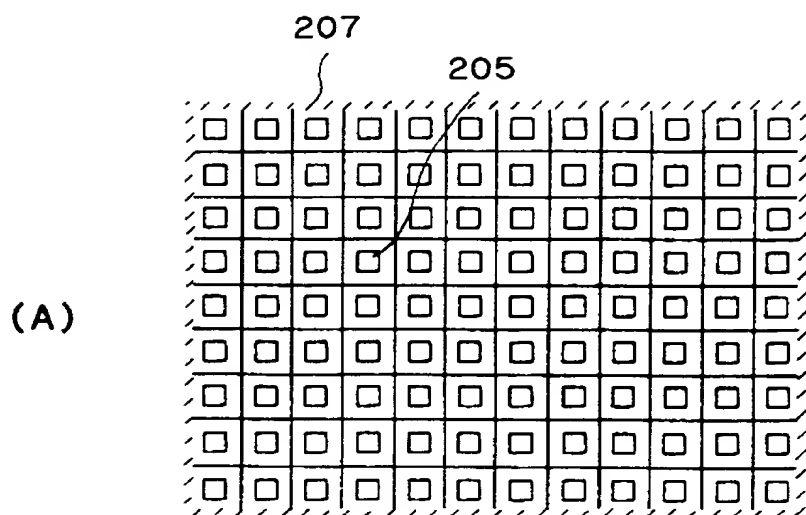
【図 3 6】



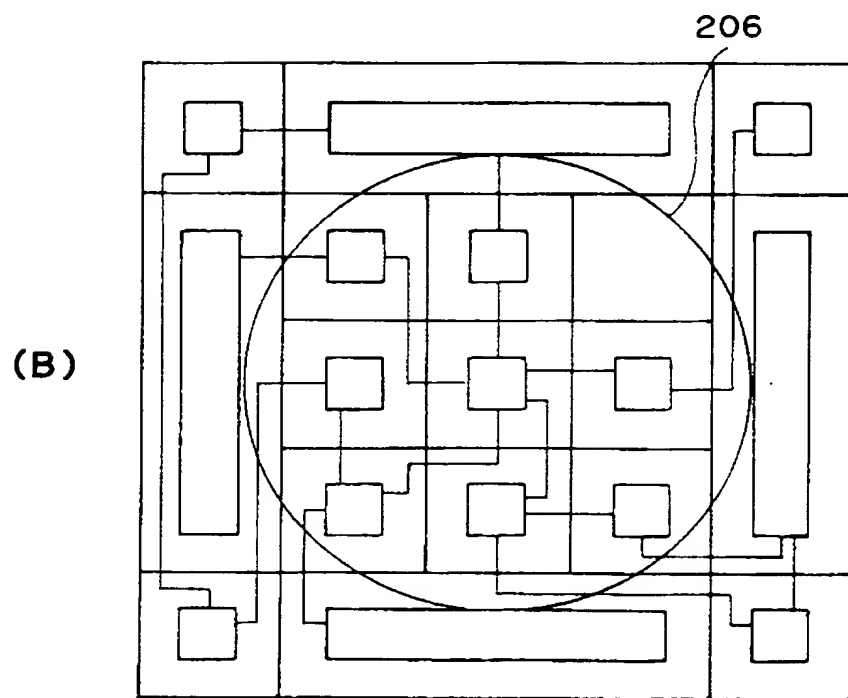
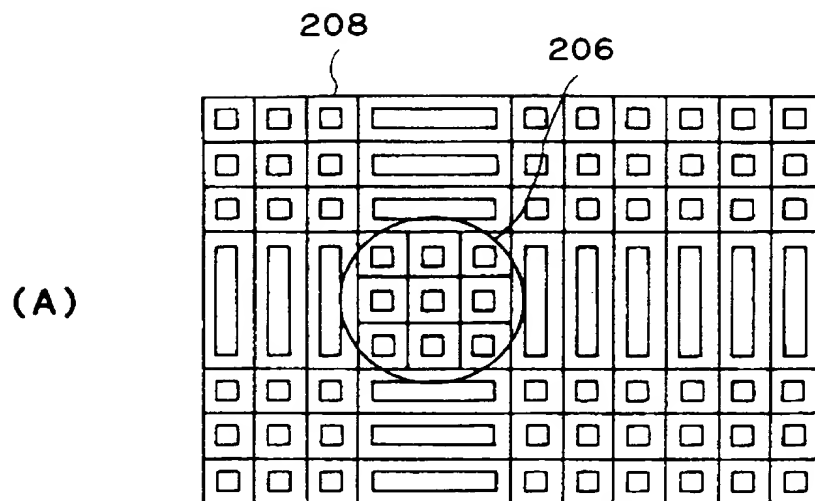
【図 3 7】



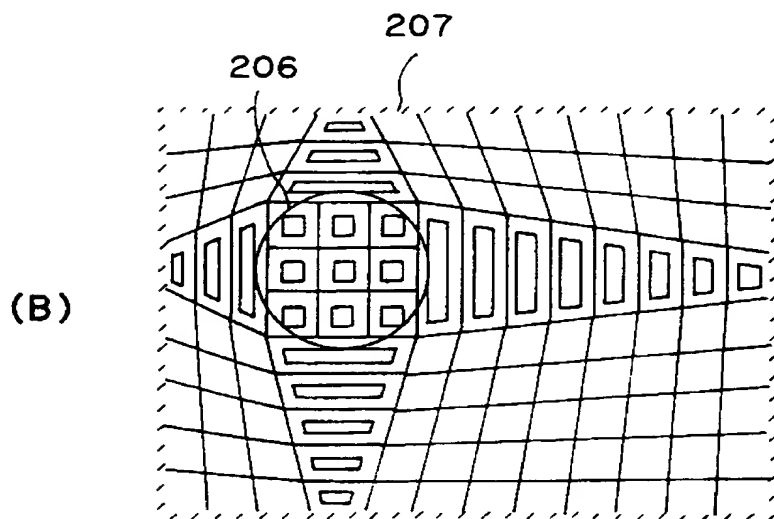
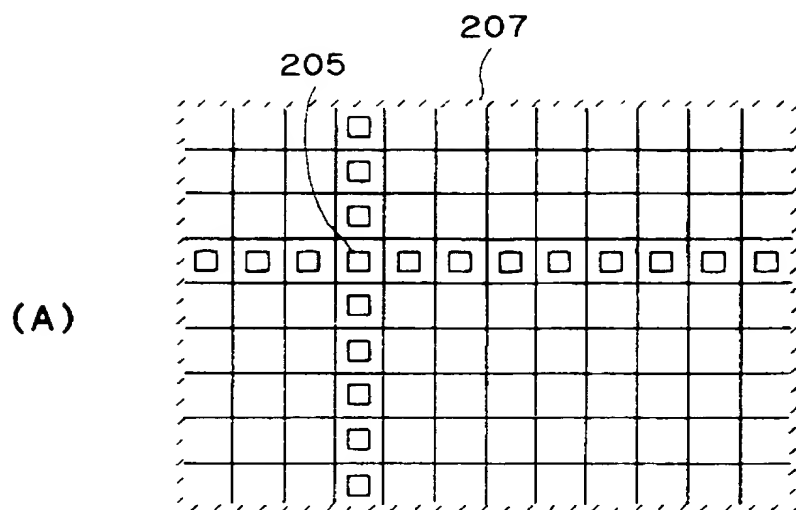
【図 3 8】



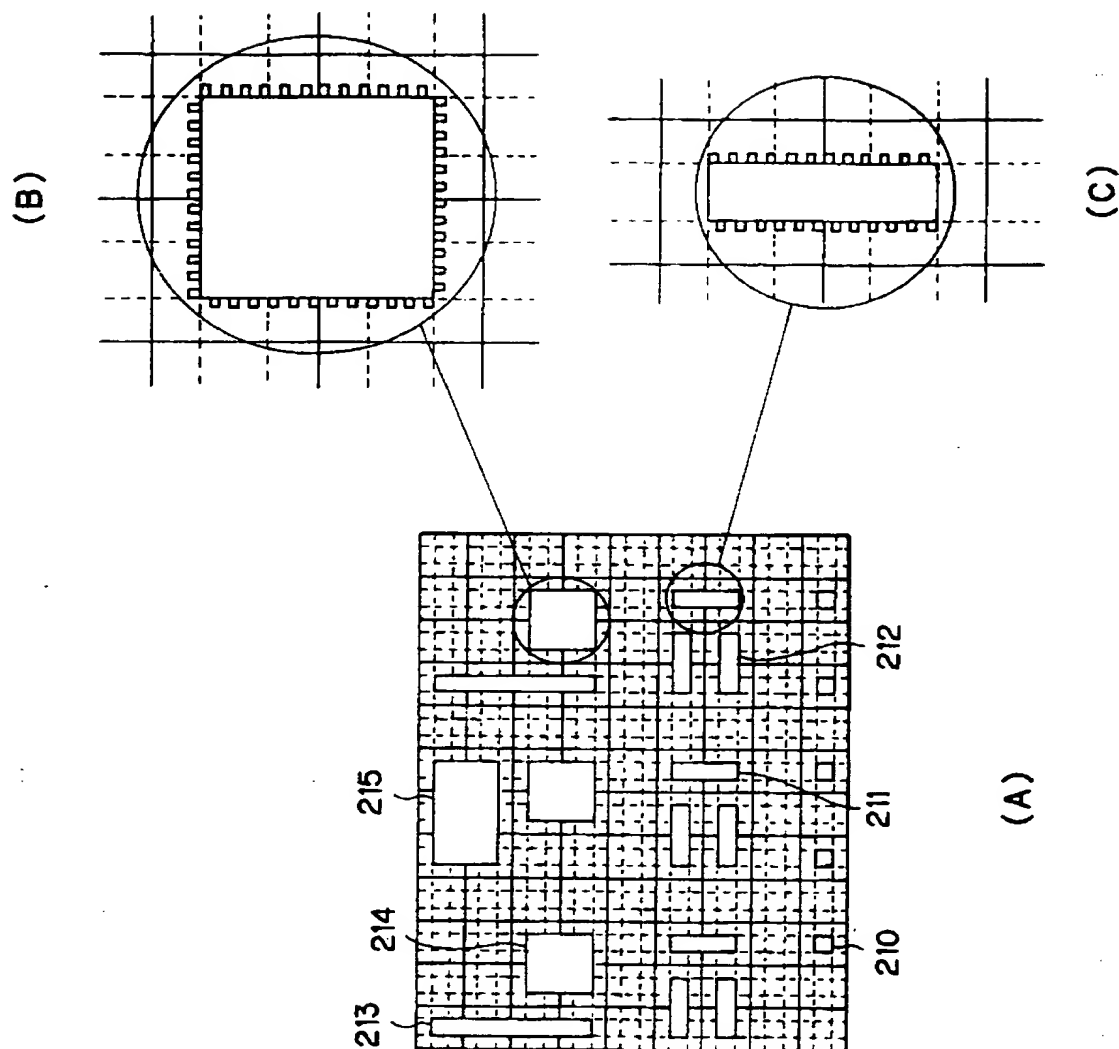
【図 3 9】



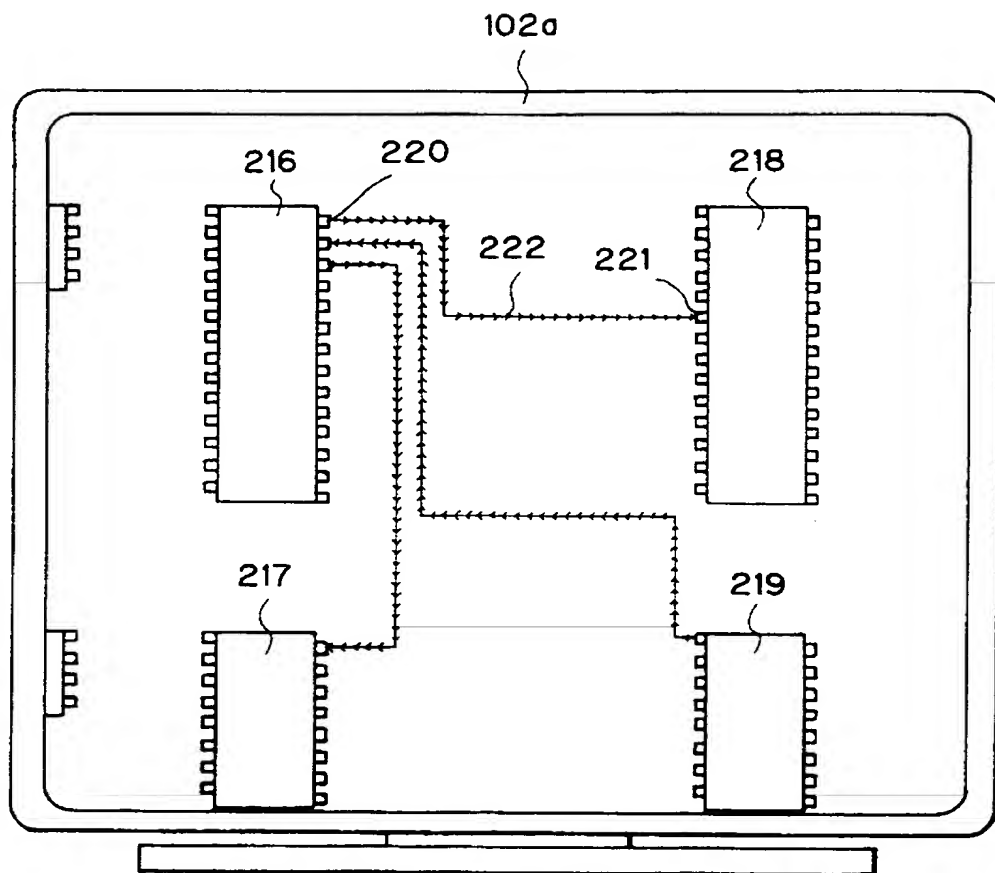
【図 4 0】



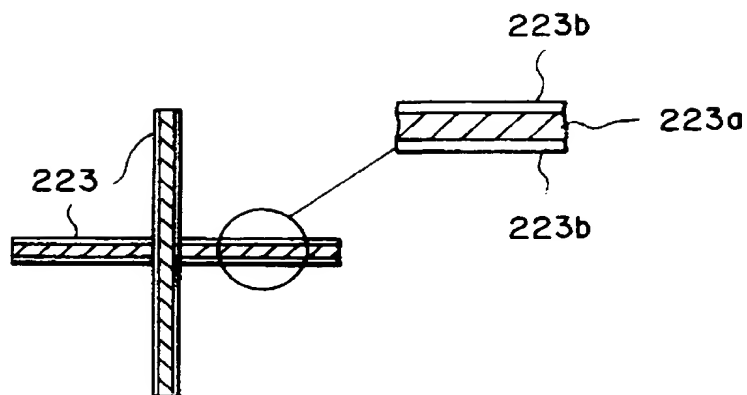
【図 4 1】



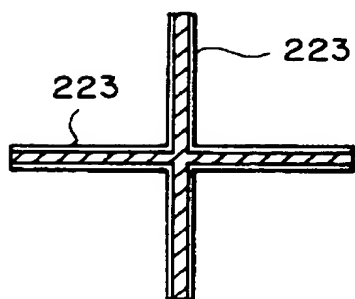
【図42】



【図 4 3】



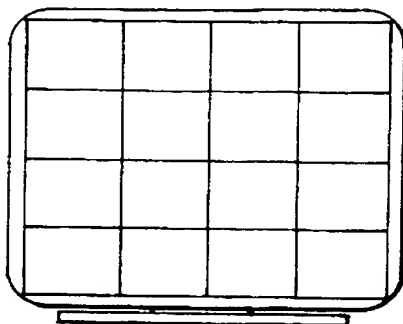
(A)



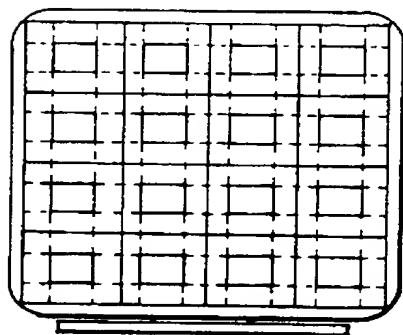
(B)

【図 4 4】

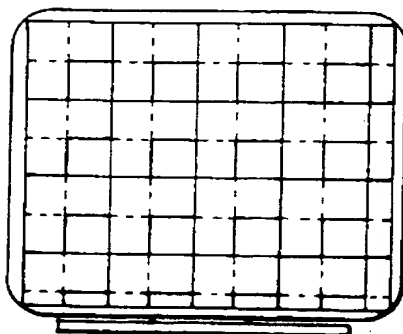
(A)



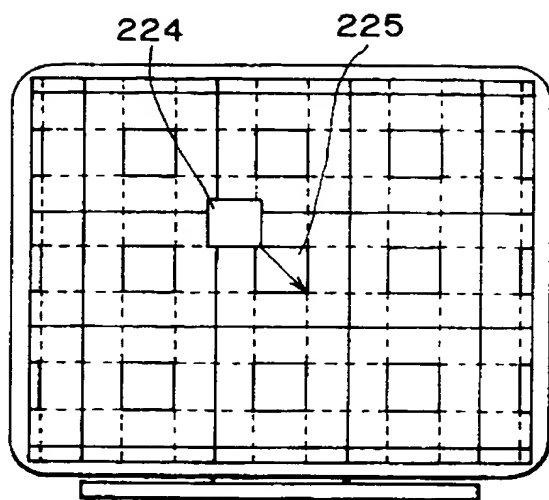
(B)



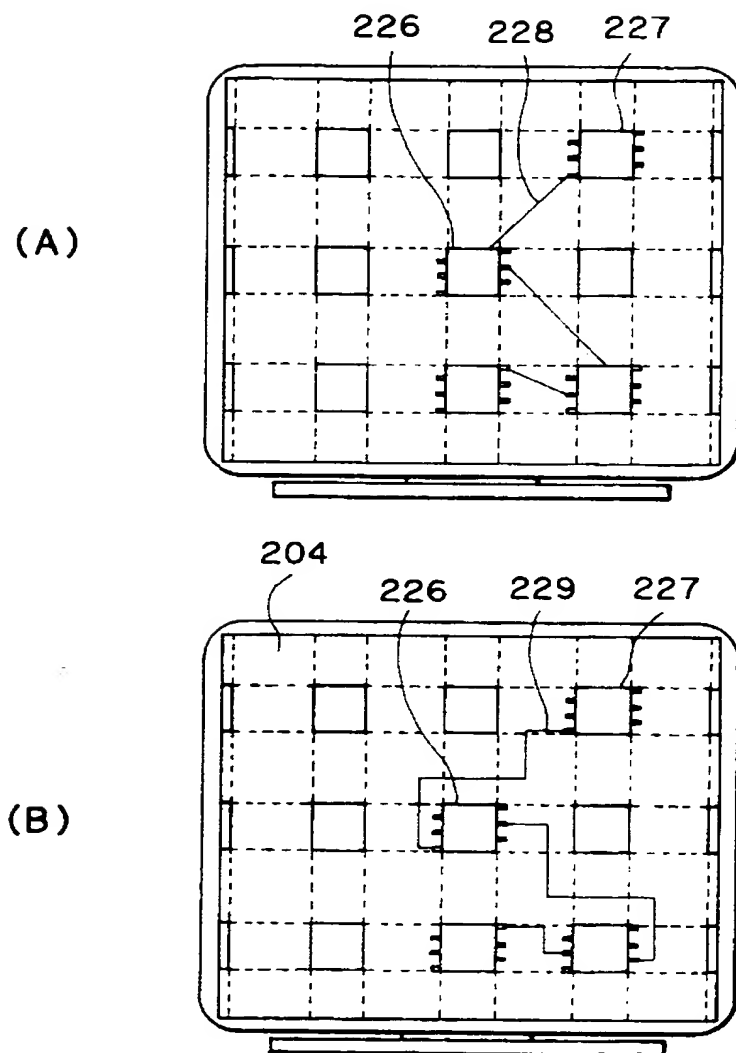
(C)



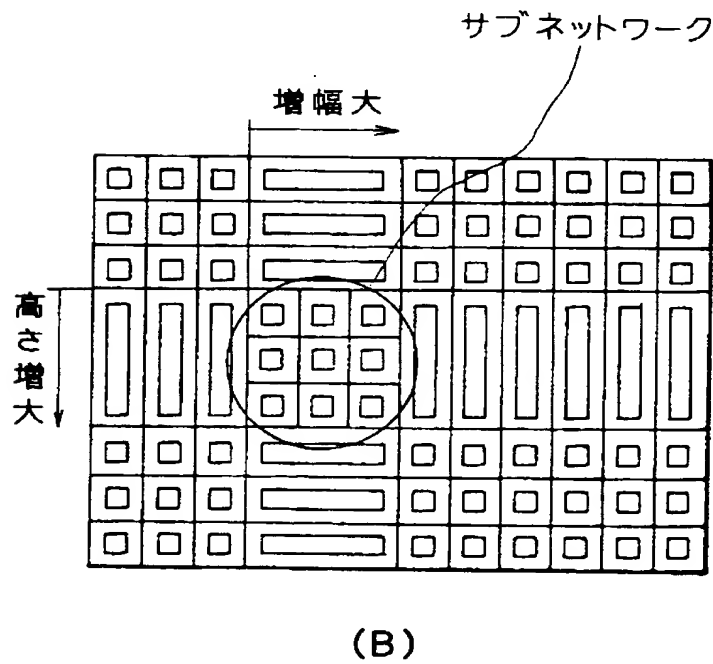
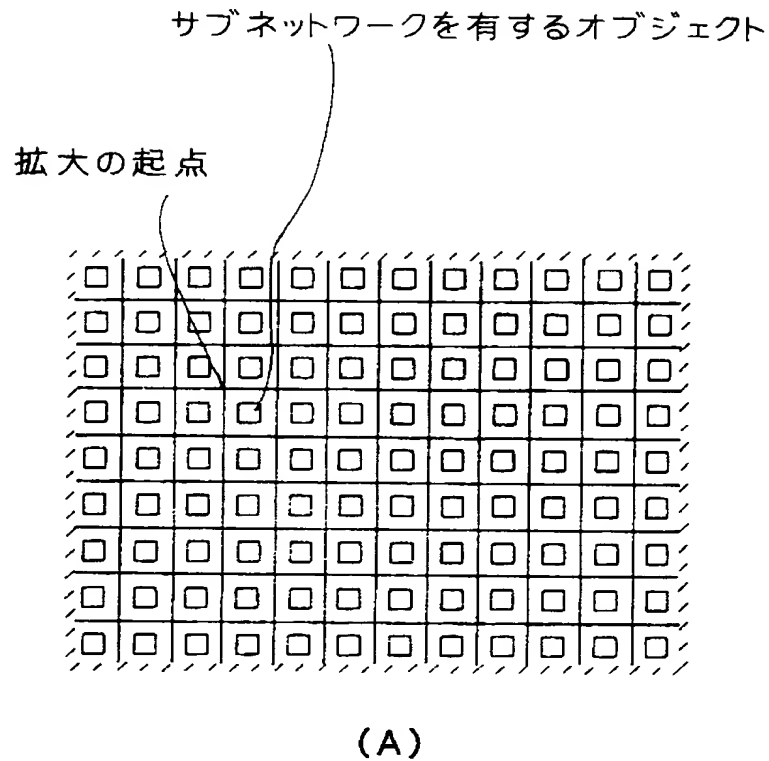
【図 4 5】



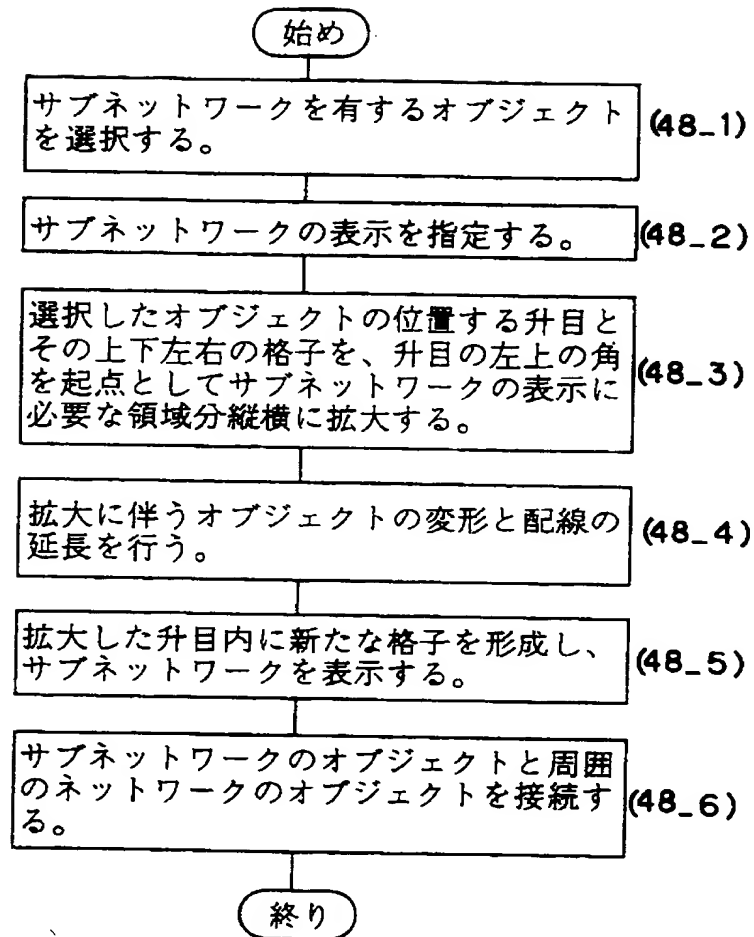
【図46】



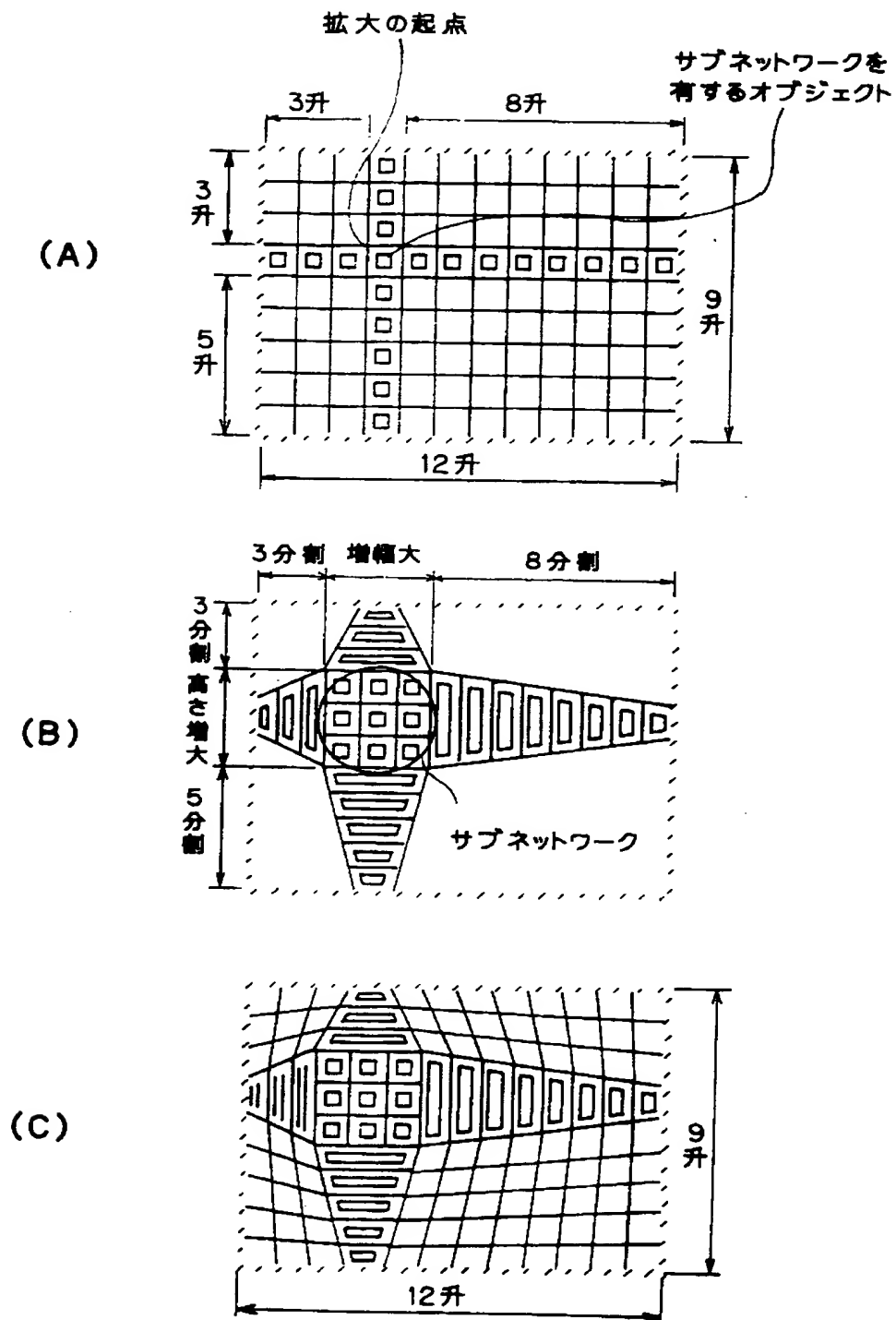
【図47】



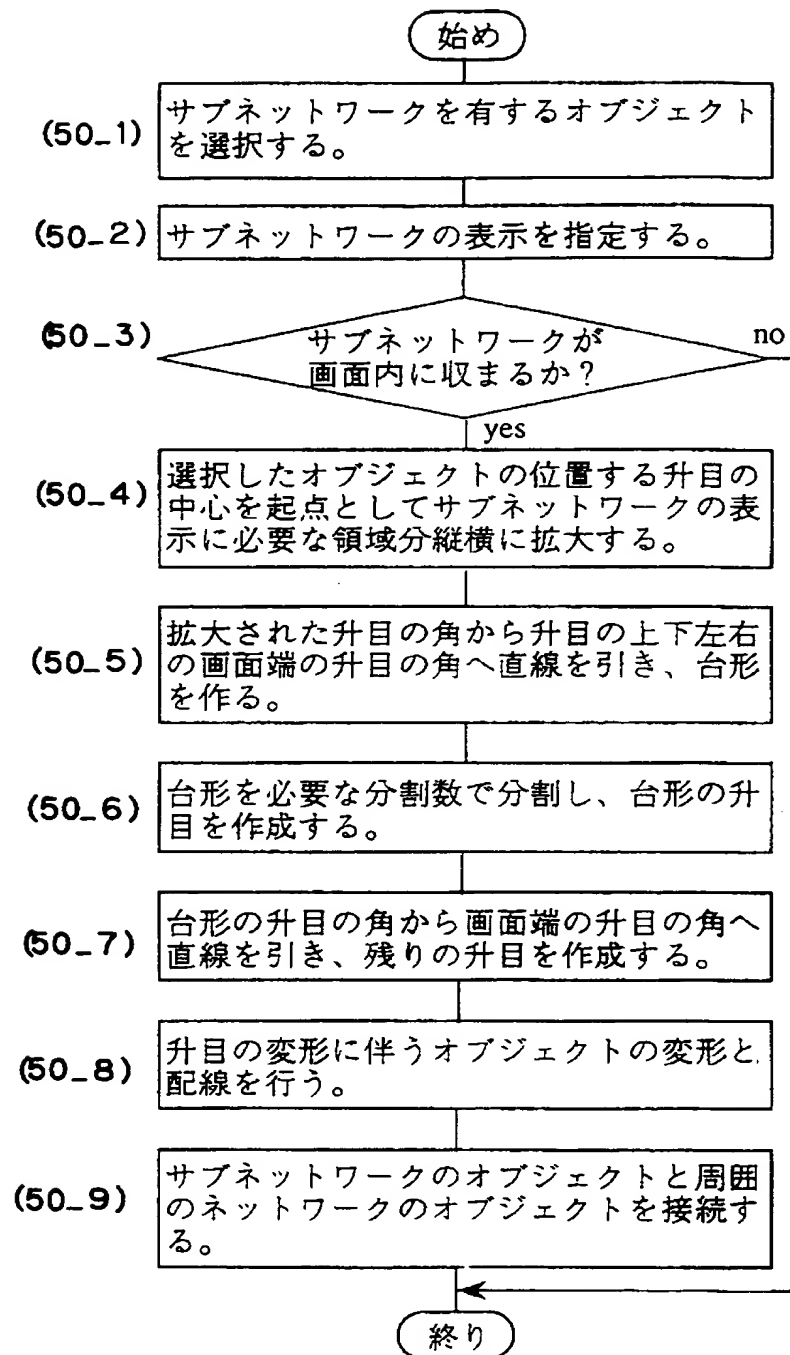
【図 48】



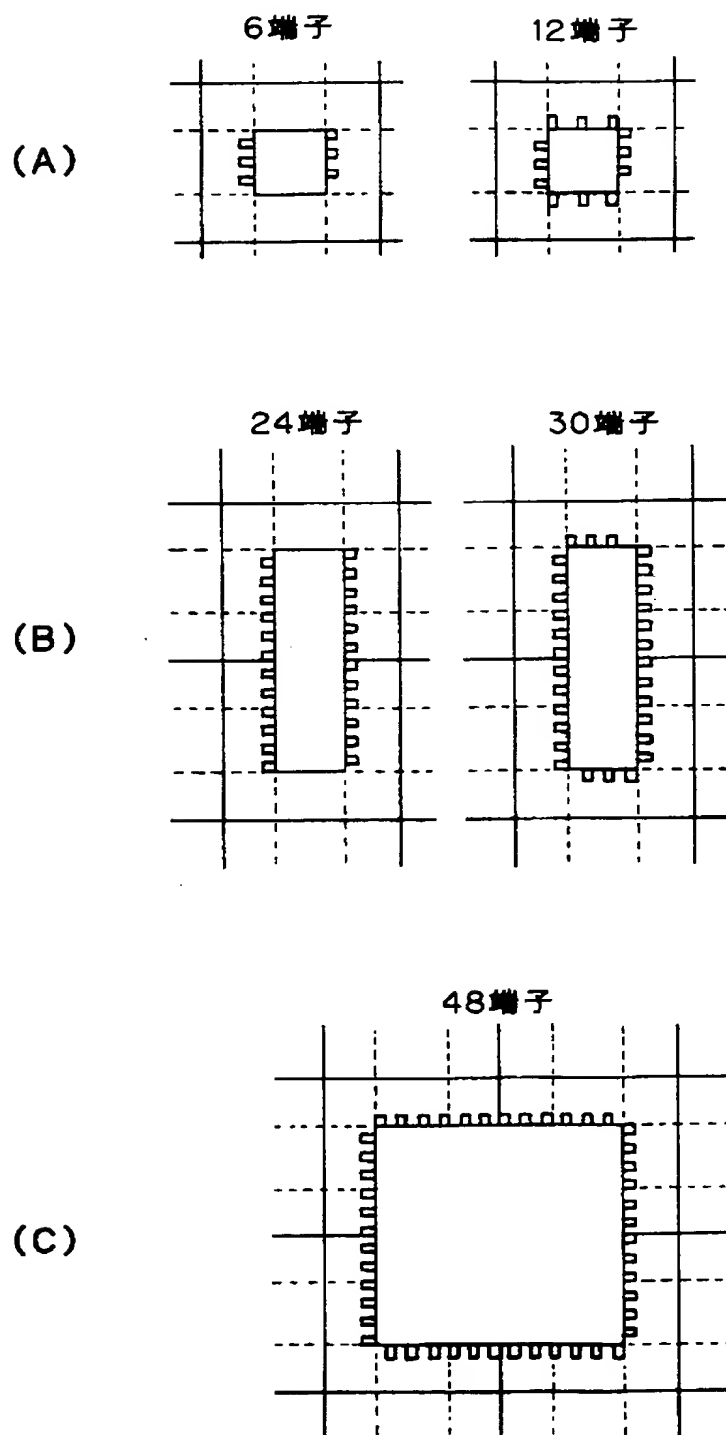
【図49】



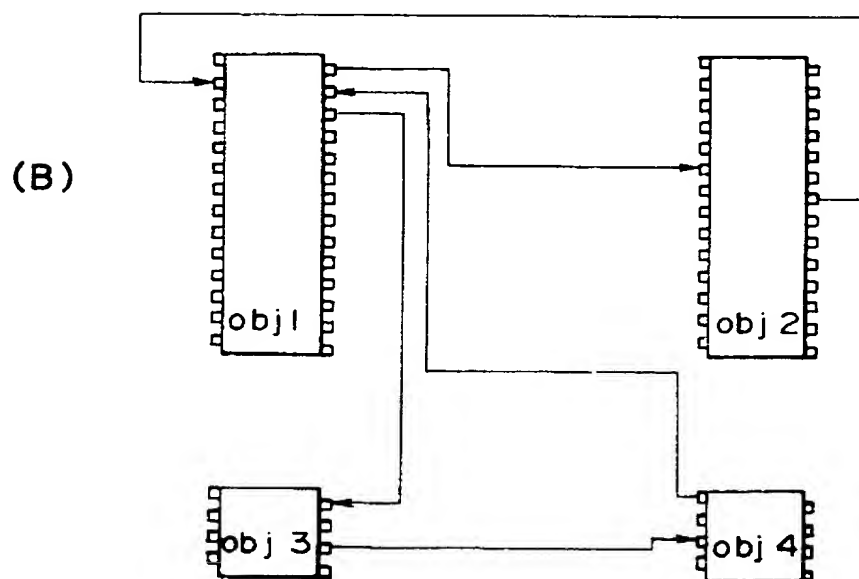
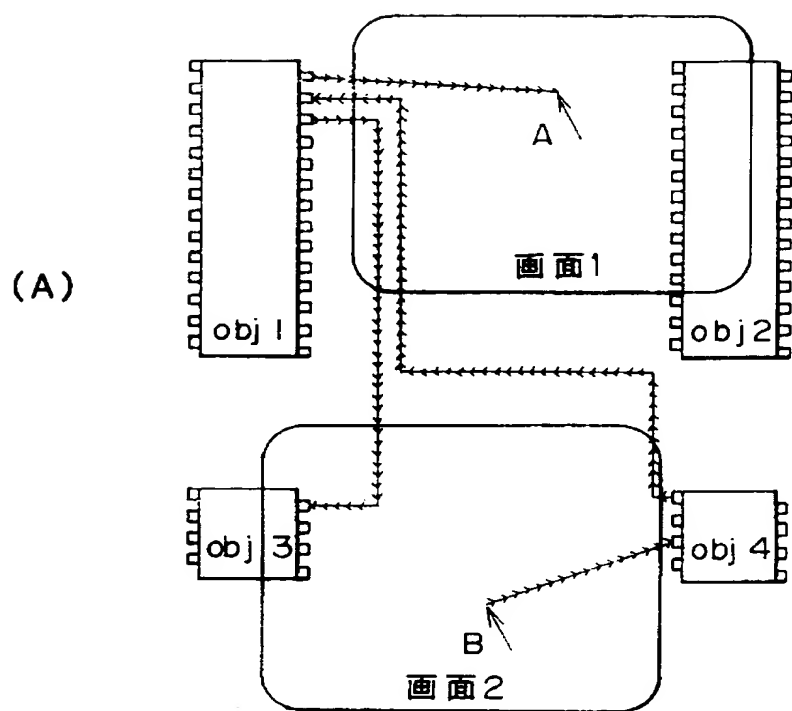
【図50】



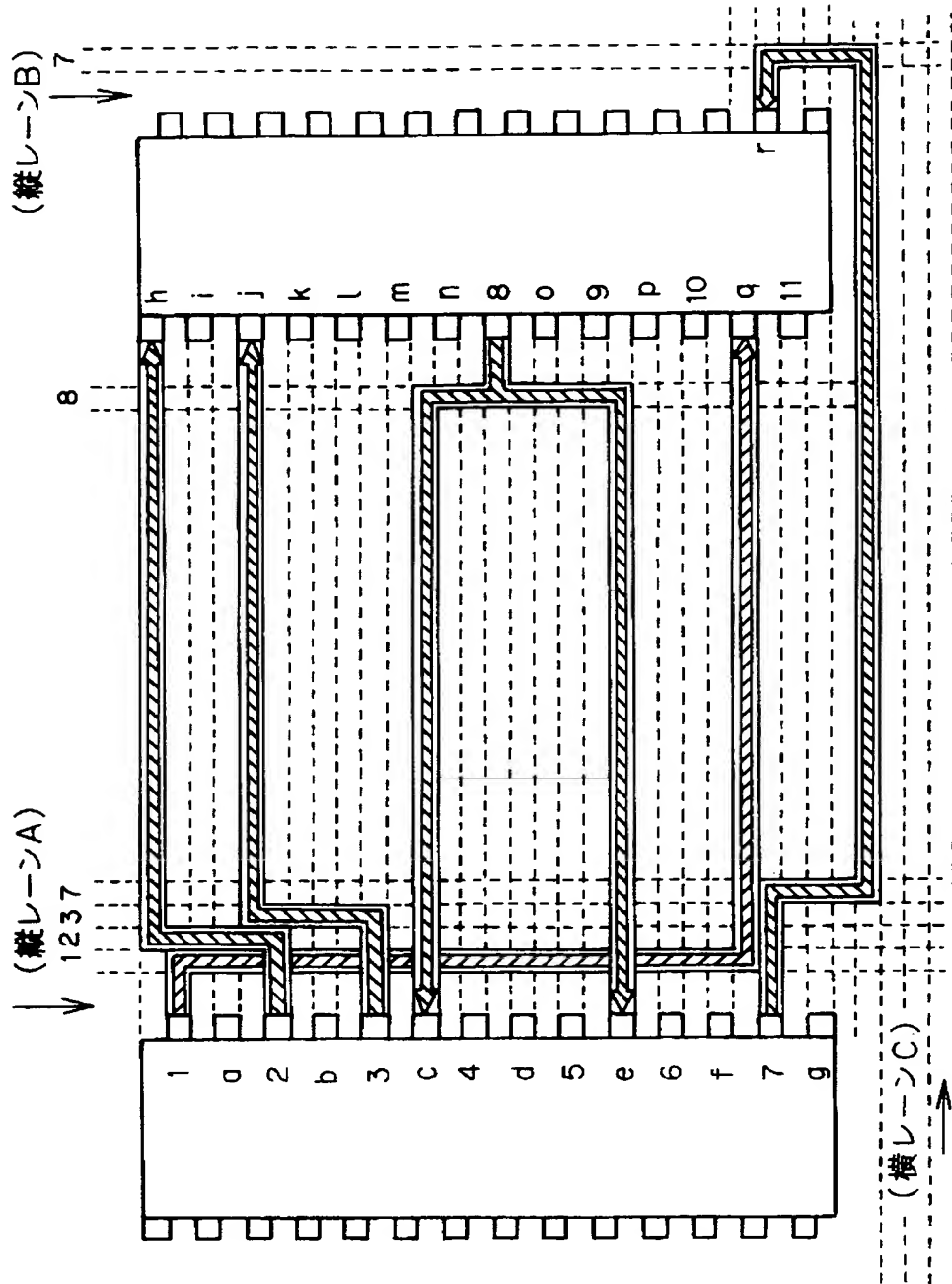
【図51】



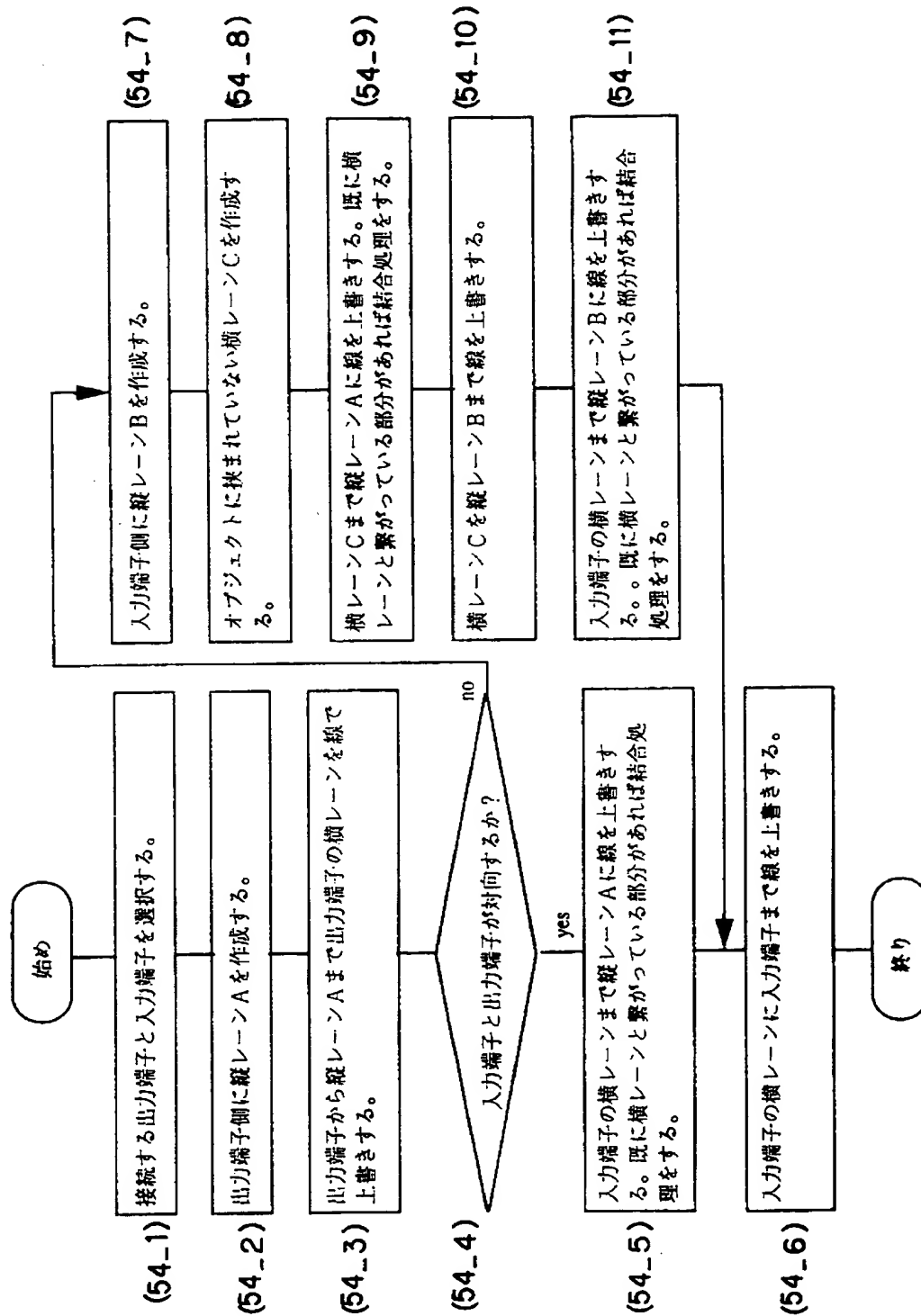
【図52】



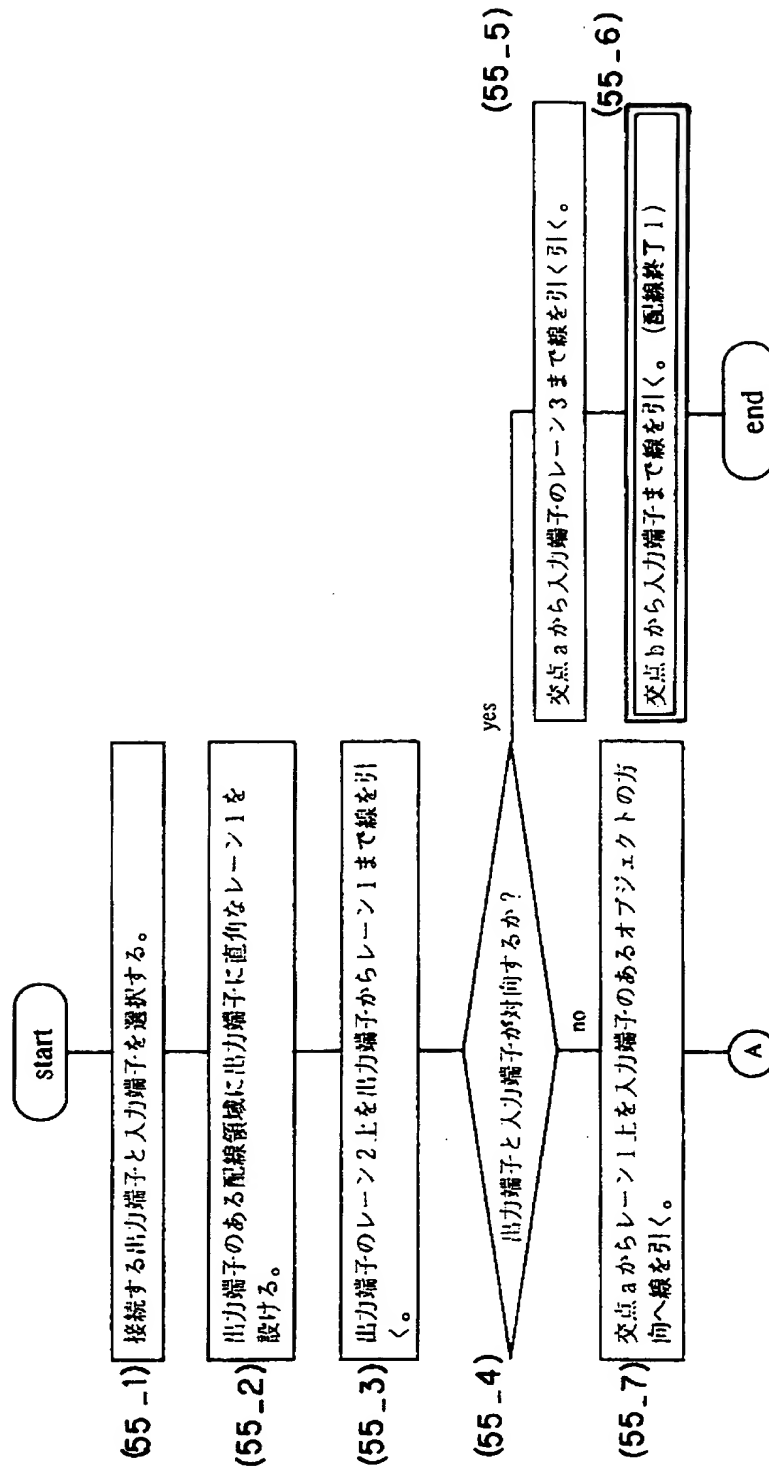
【図53】



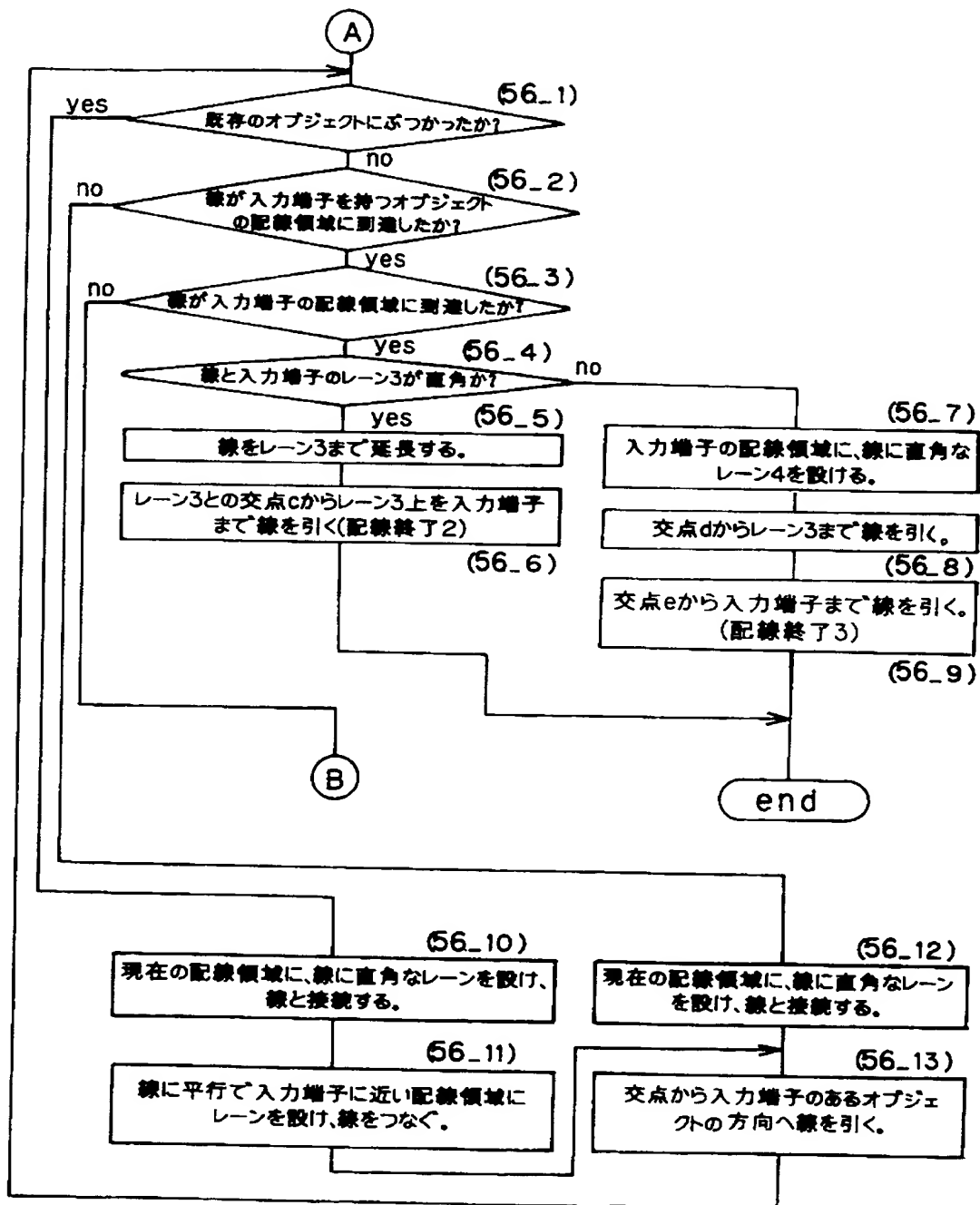
【図54】



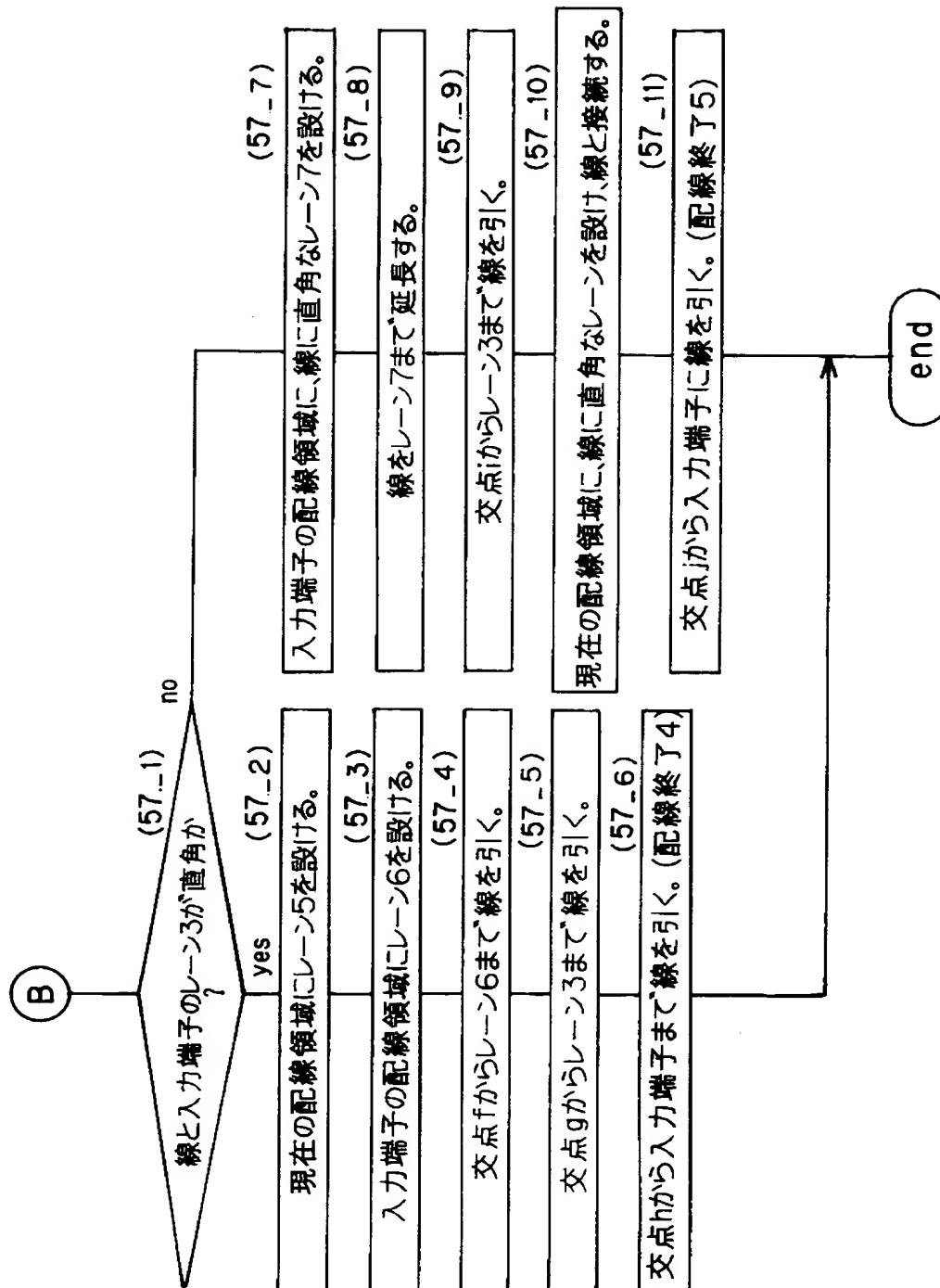
【図55】



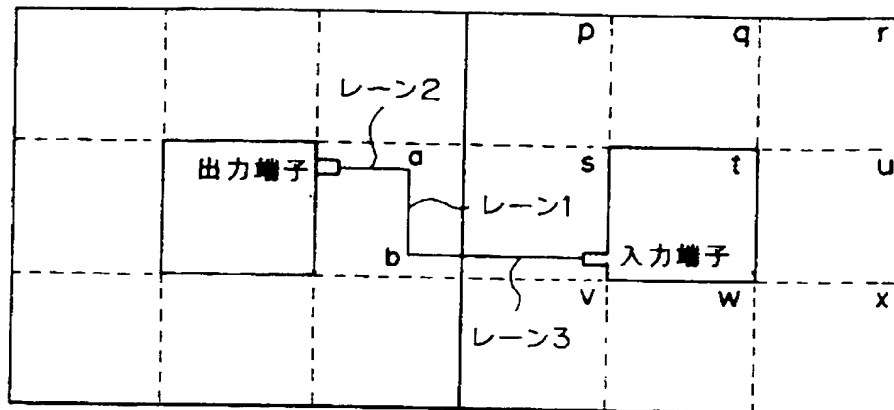
【図56】



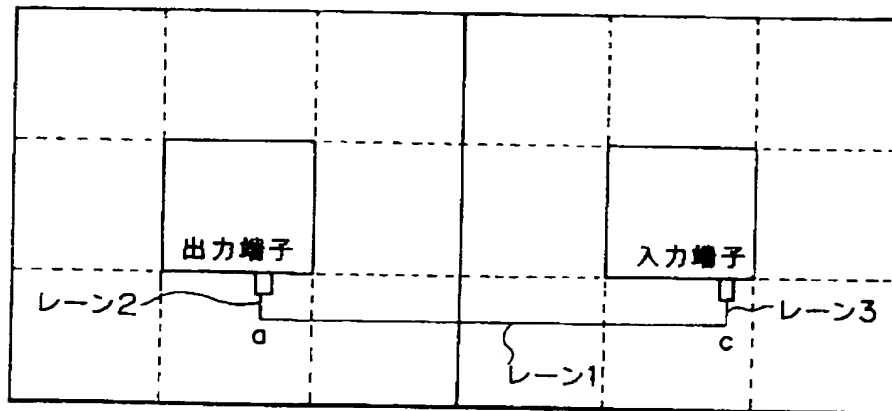
【図57】



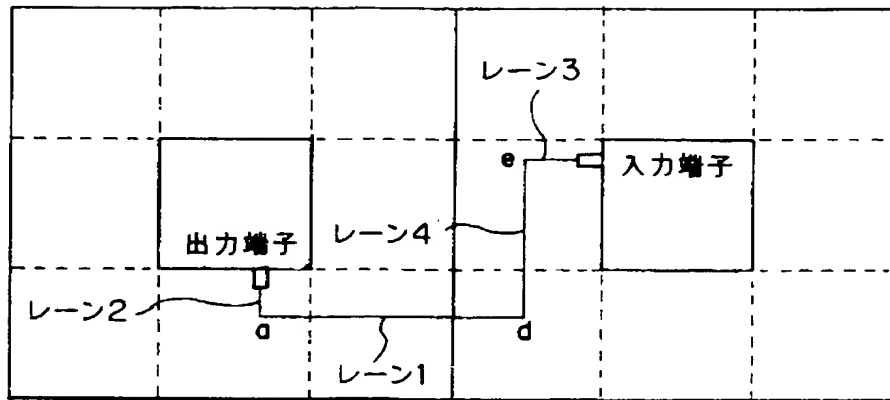
【図58】



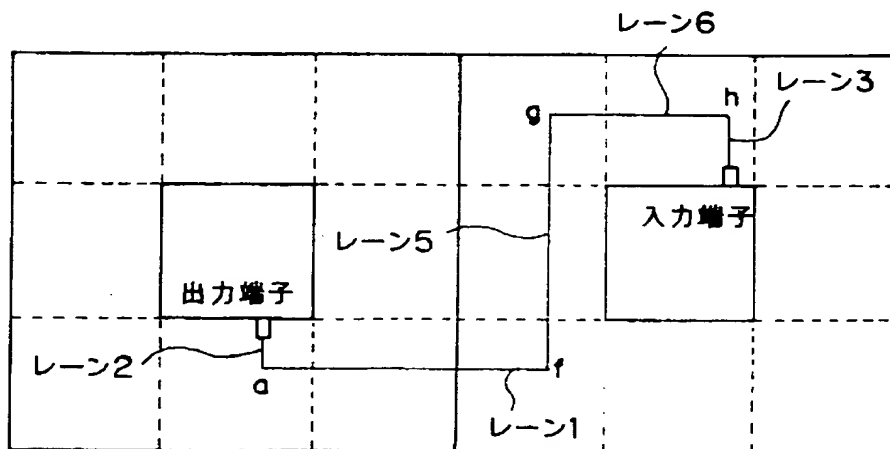
【図59】



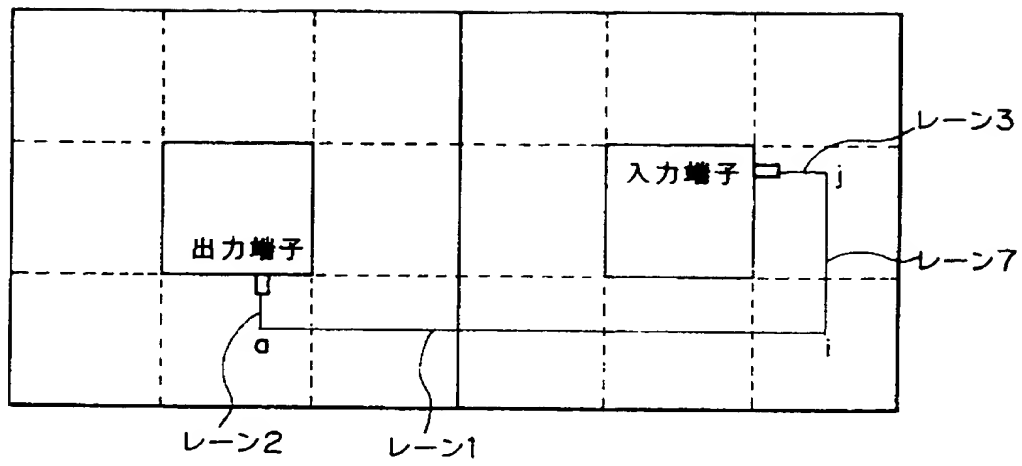
【図60】



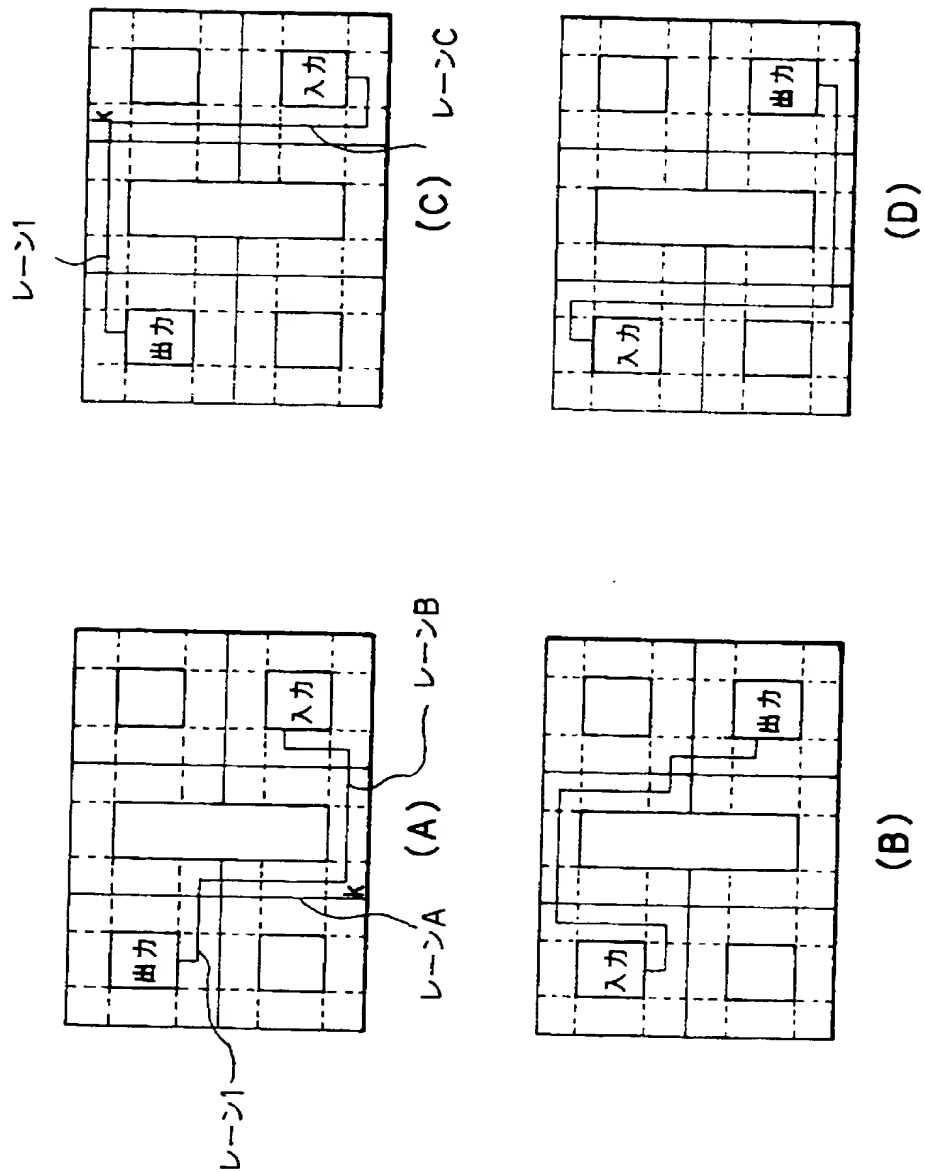
【図61】



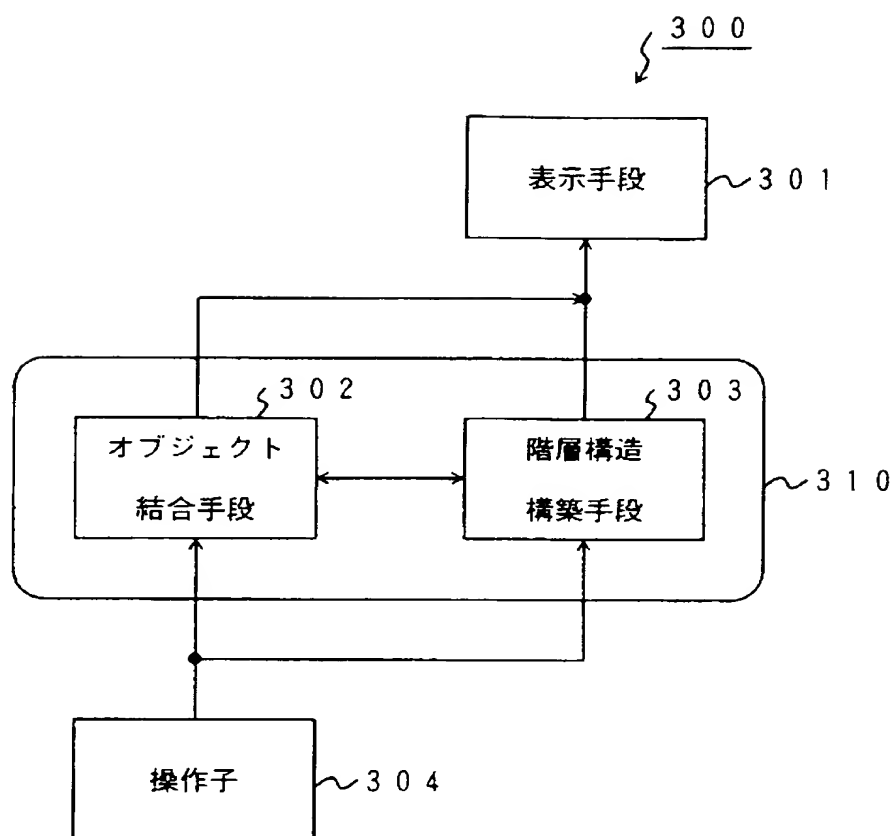
【図62】



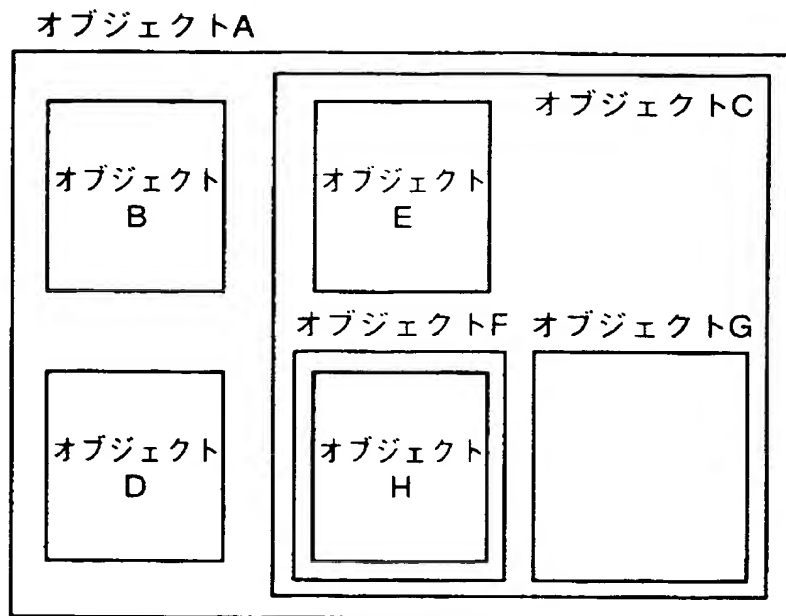
【図 6 3】



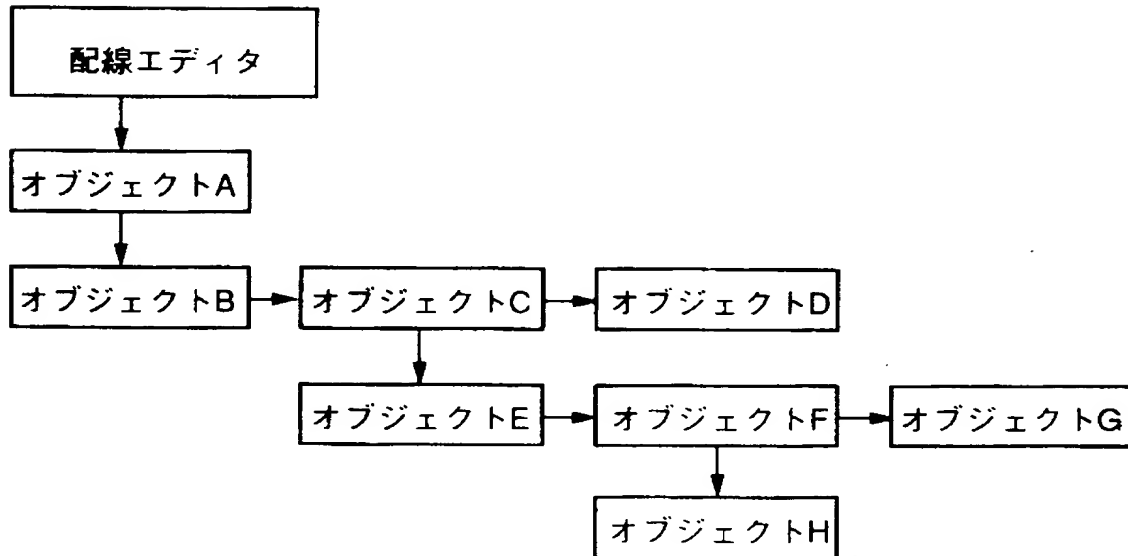
【図64】



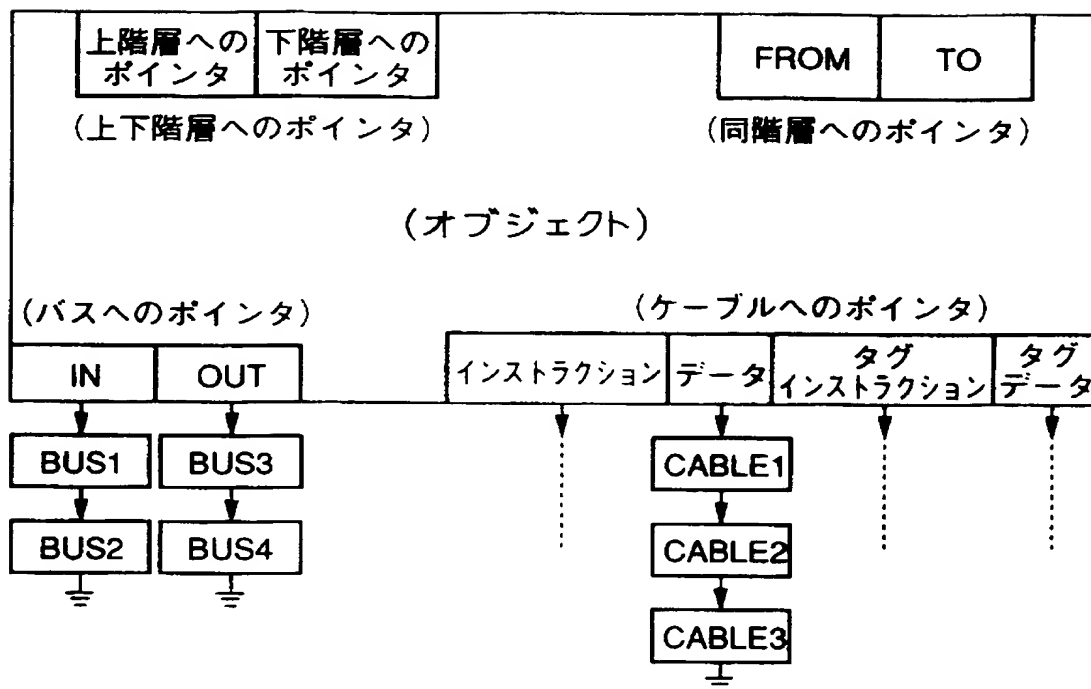
【図65】



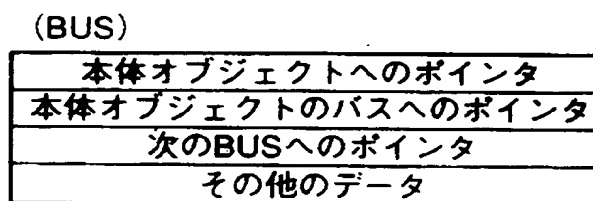
【図66】



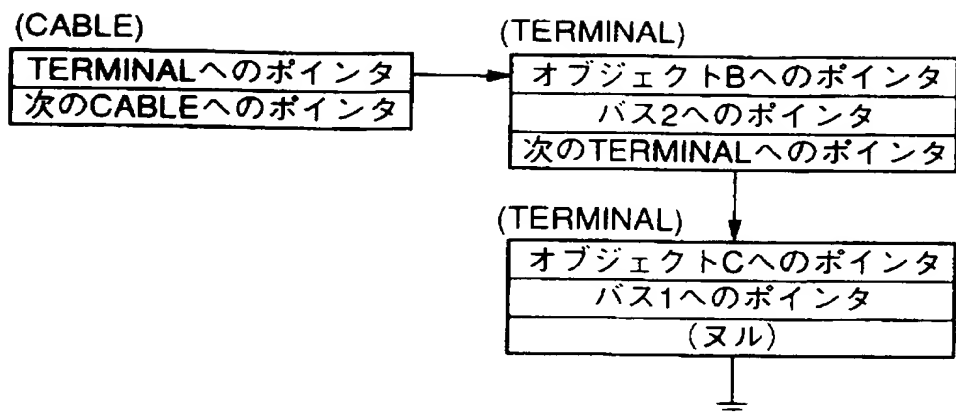
【図67】



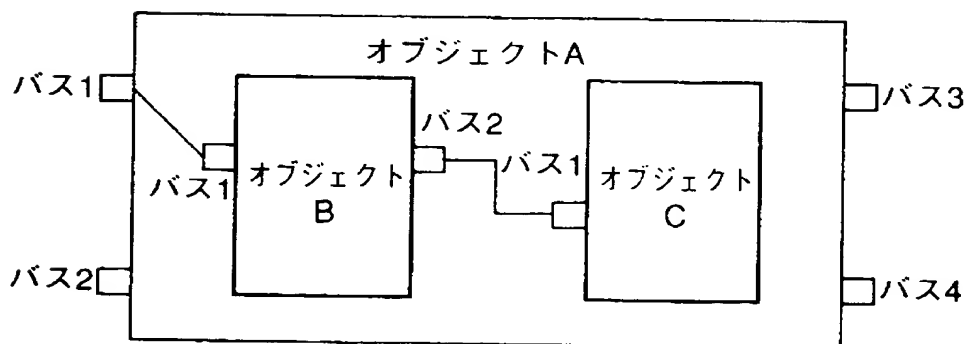
【図68】



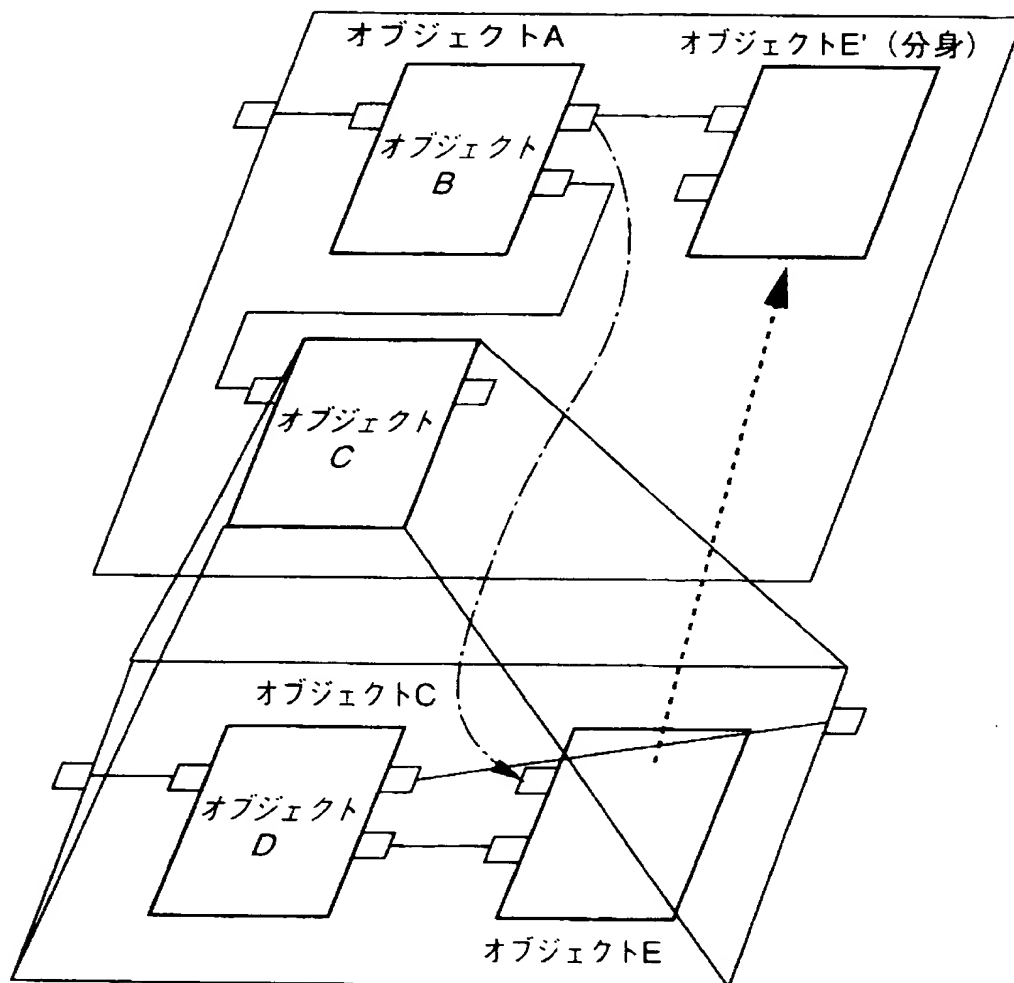
【図69】



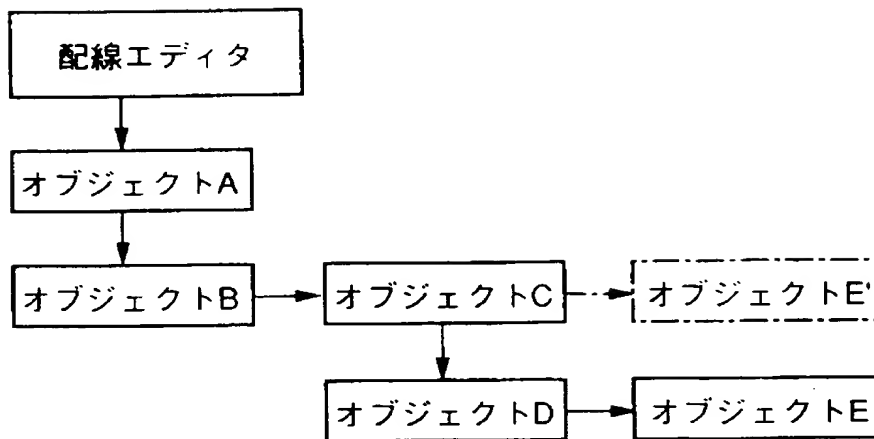
【図70】



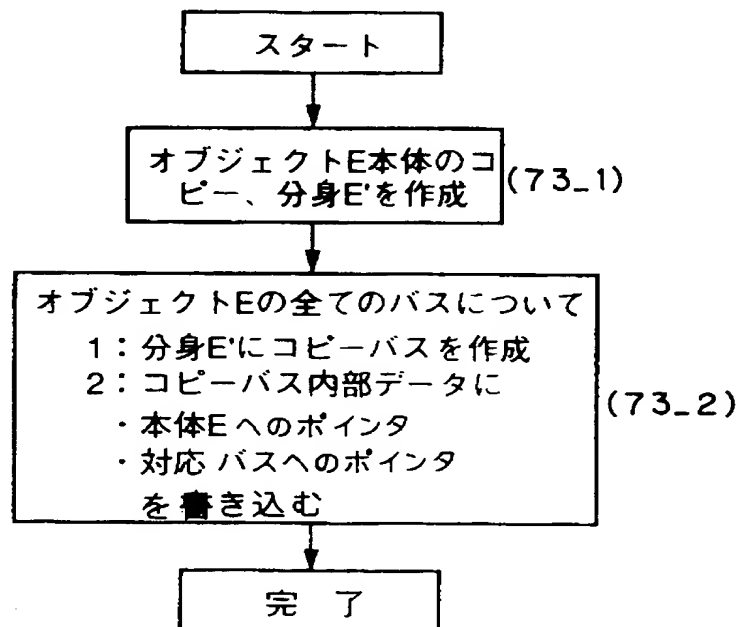
【図71】



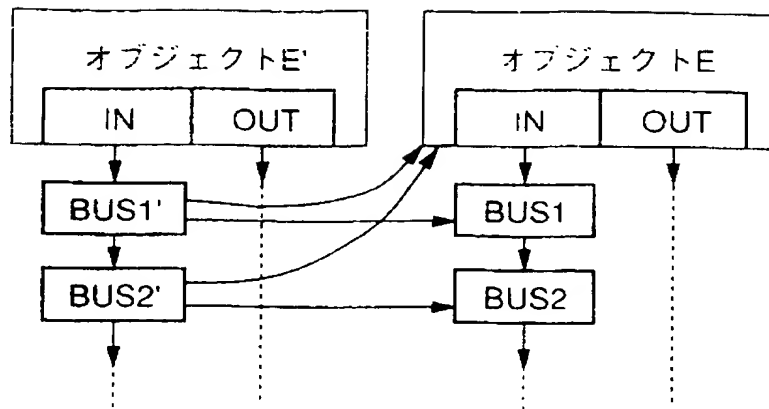
【図72】



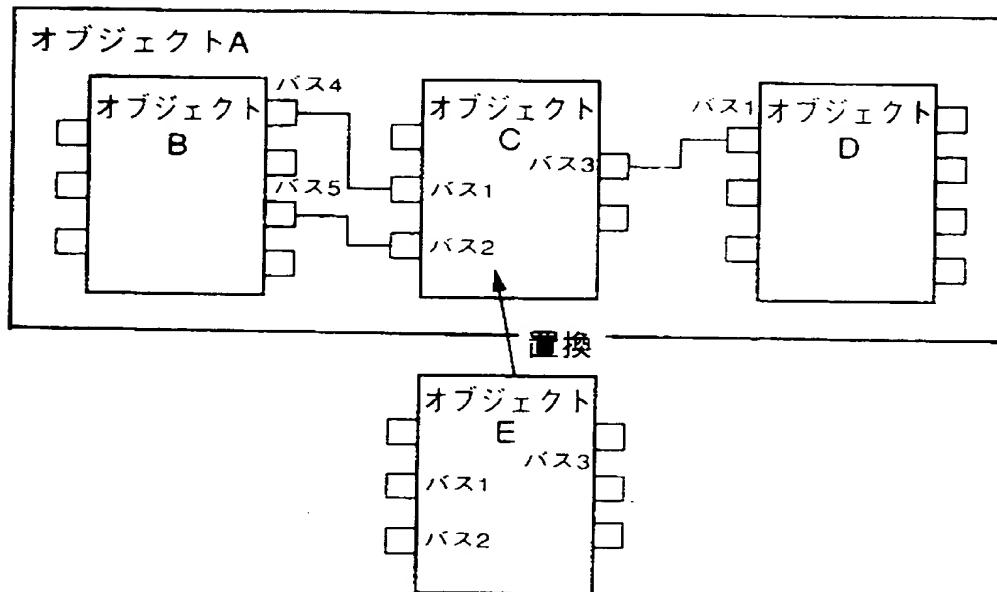
【図73】



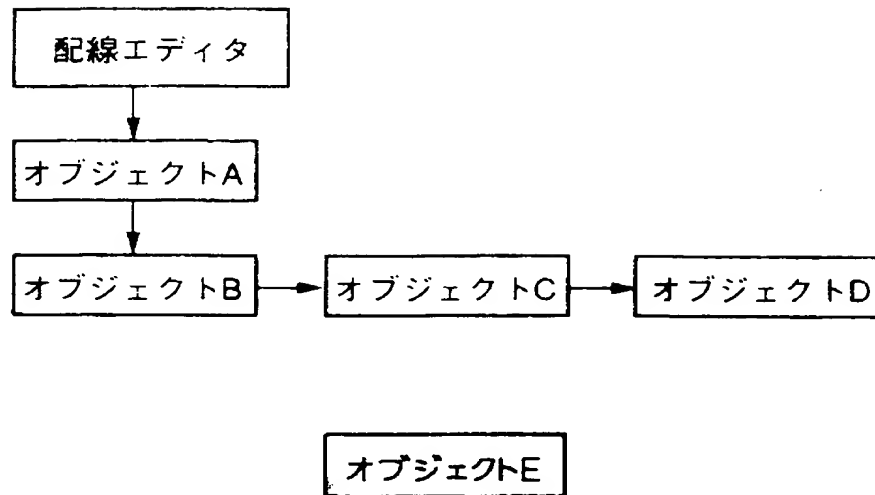
【図74】



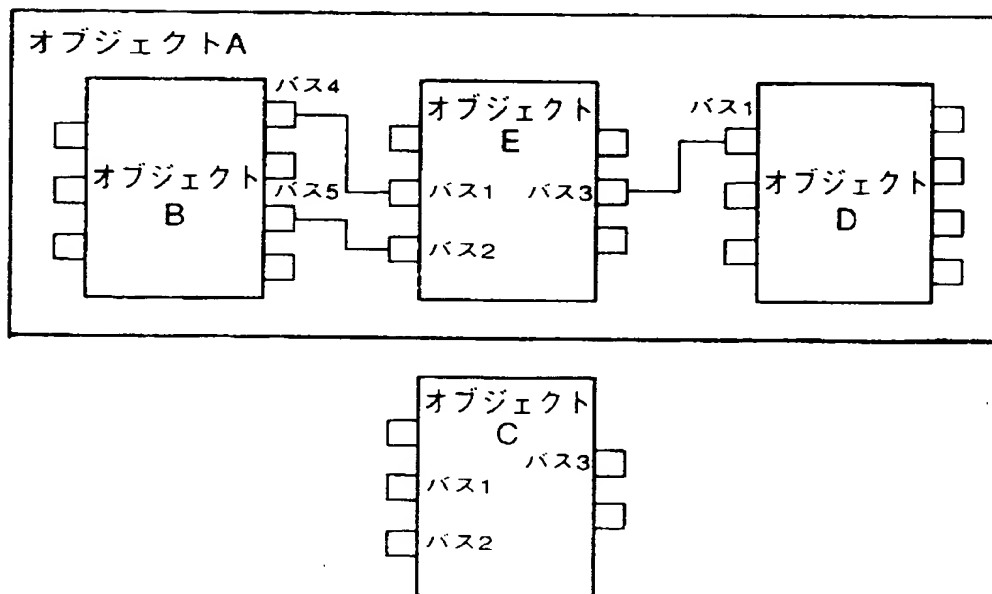
【図75】



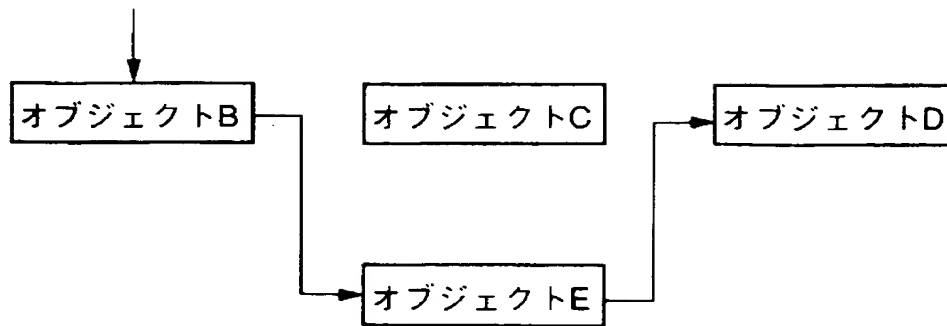
【図76】



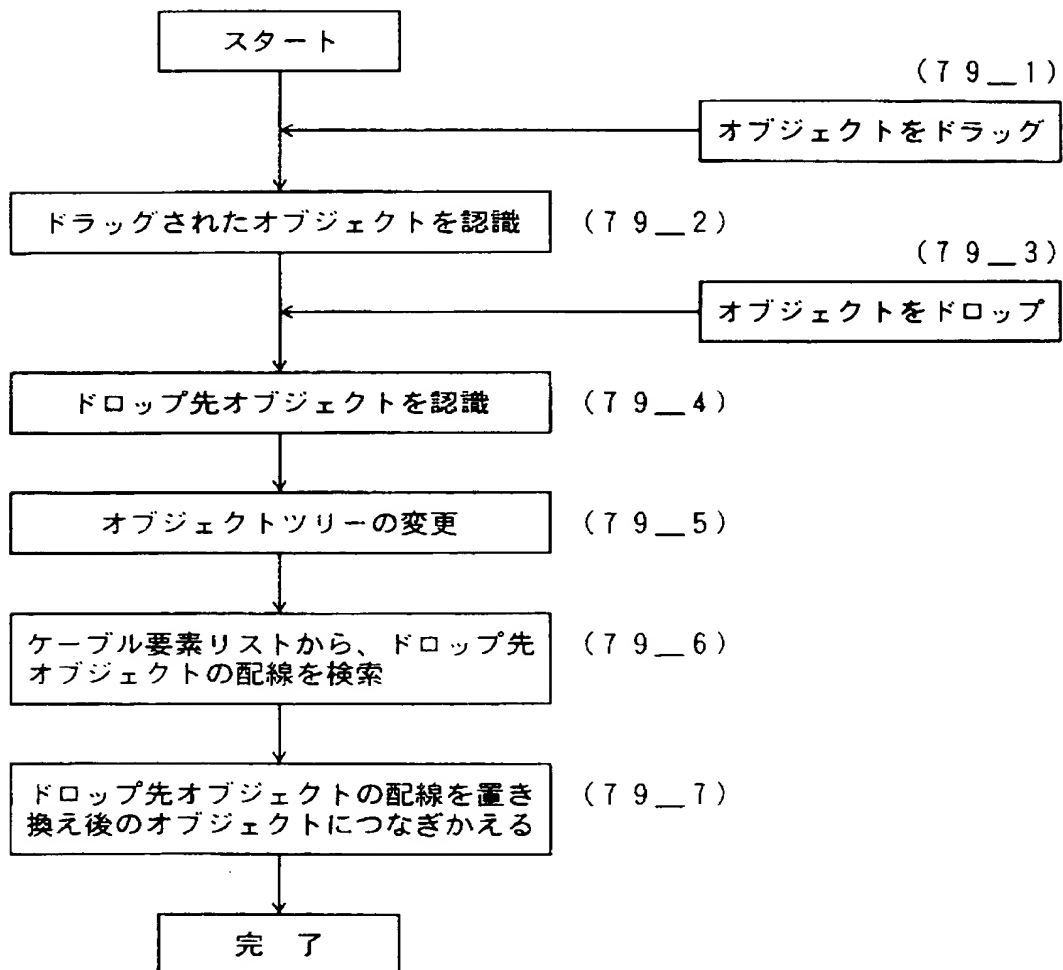
【図77】



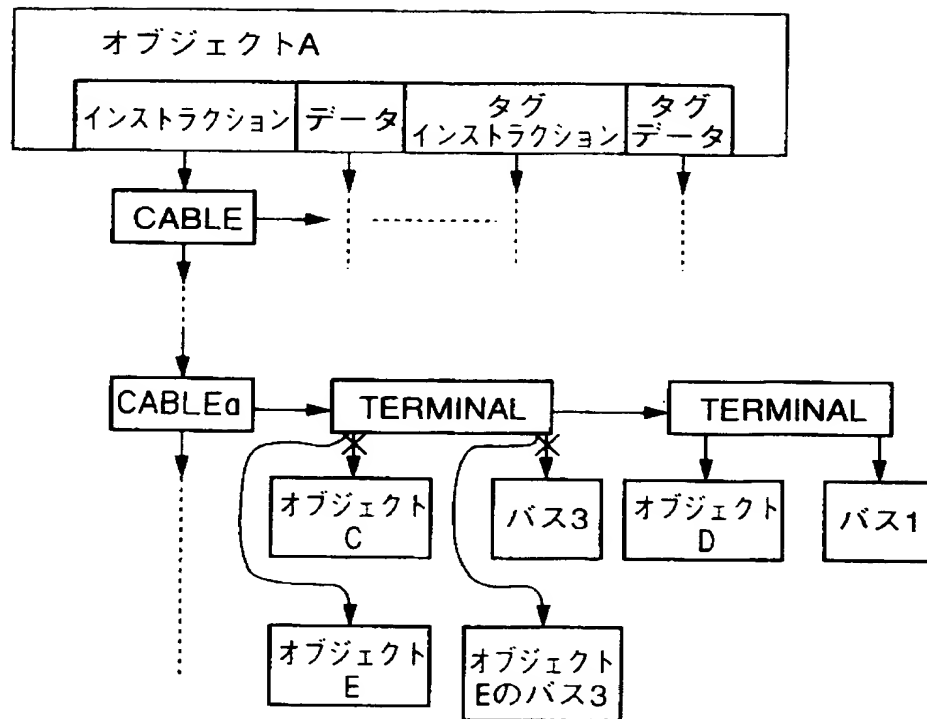
【図78】



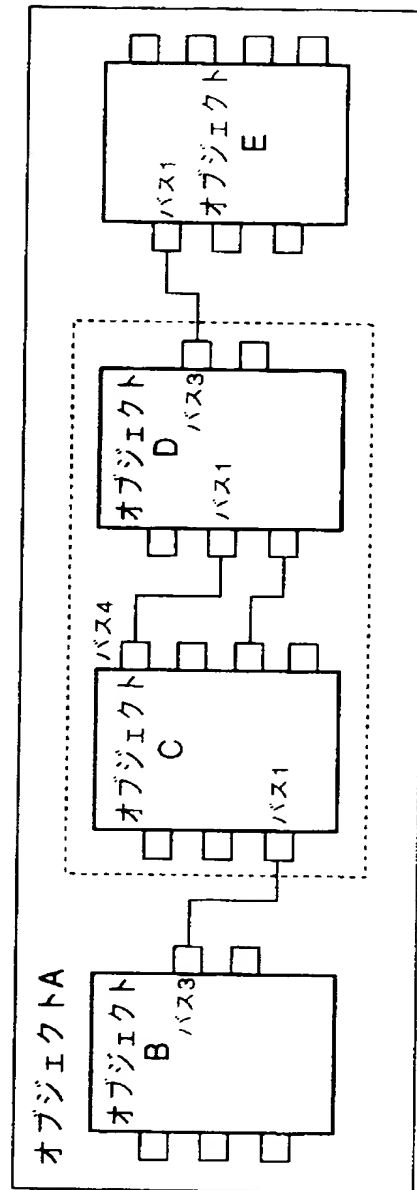
【図79】



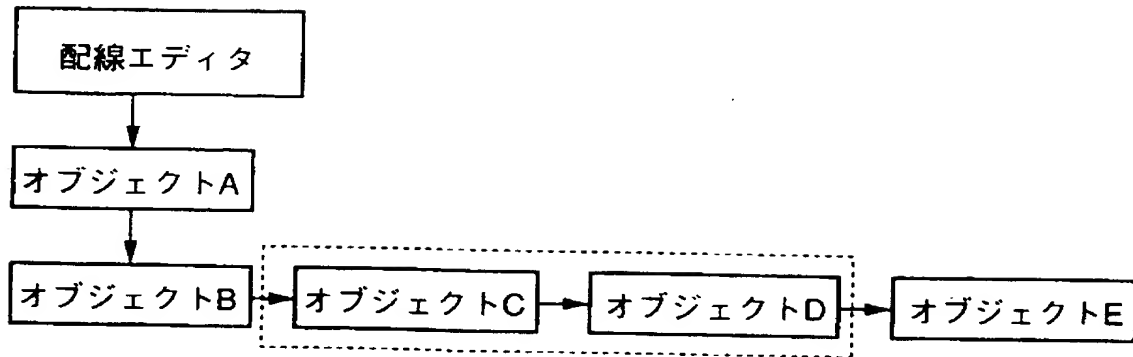
【図80】



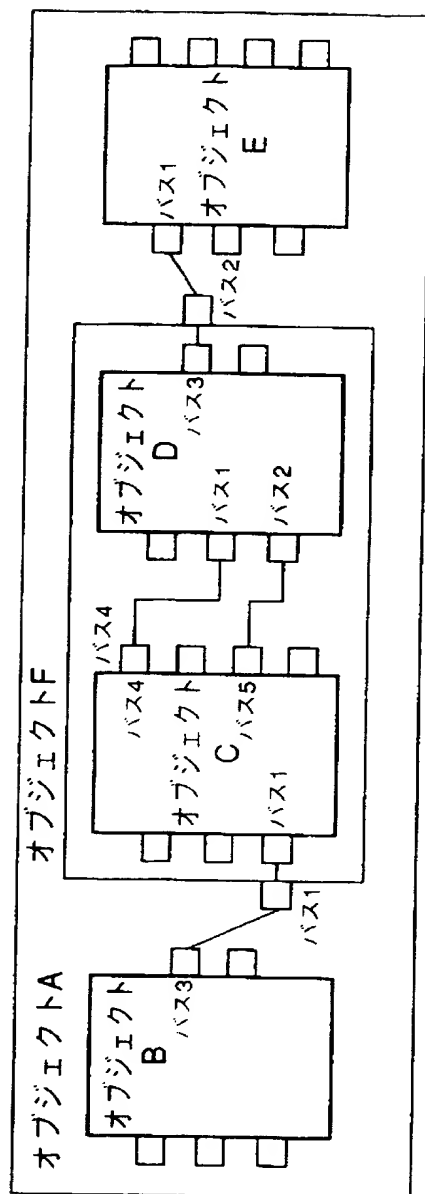
【図81】



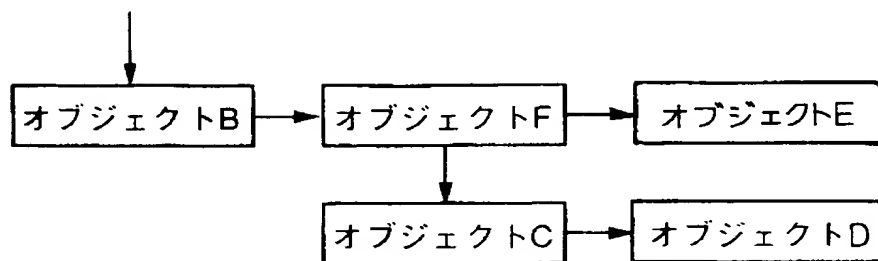
【図 82】



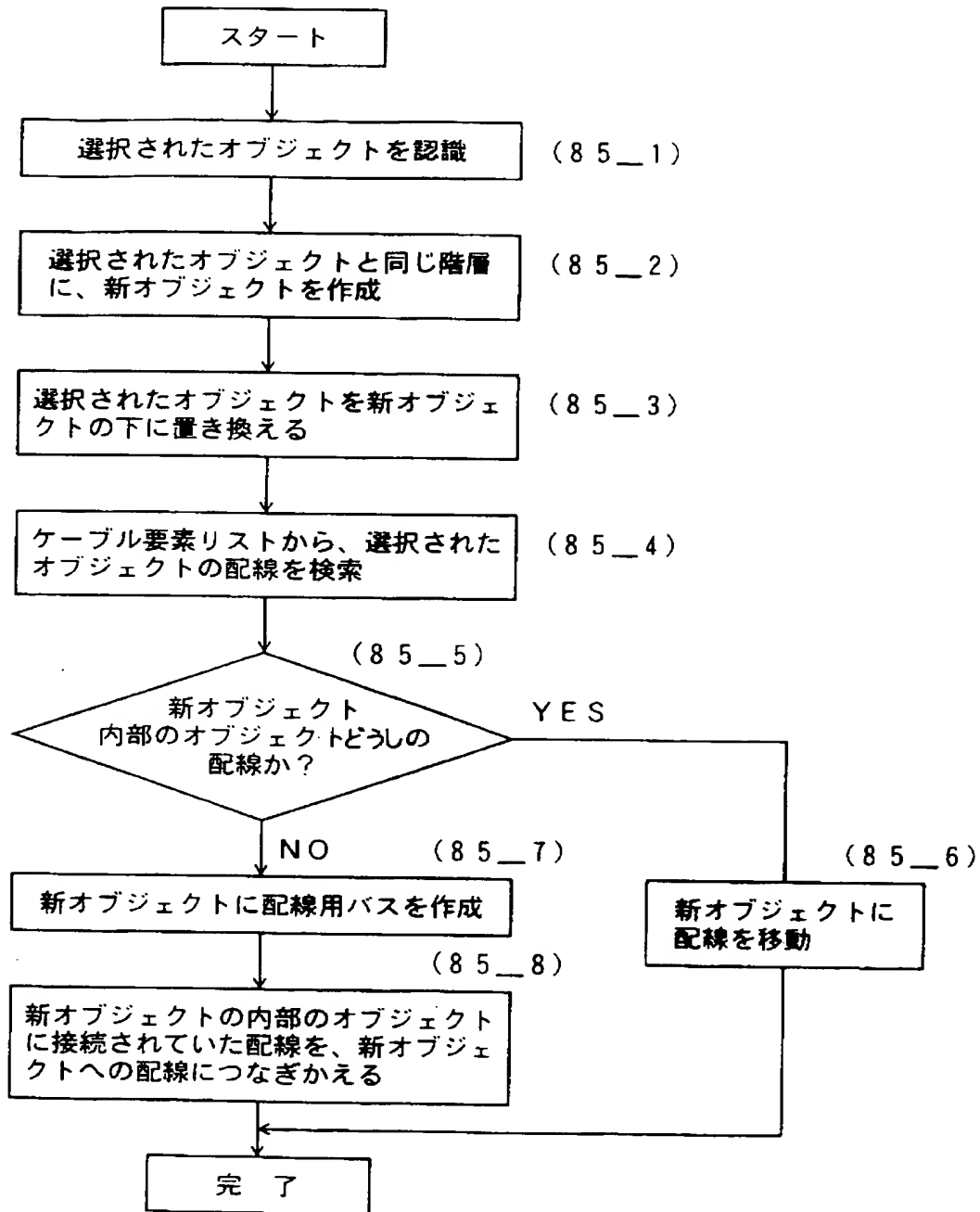
【図83】



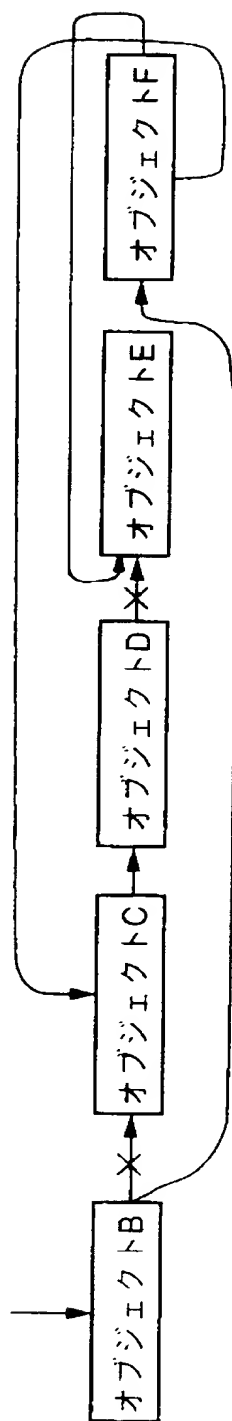
【図84】



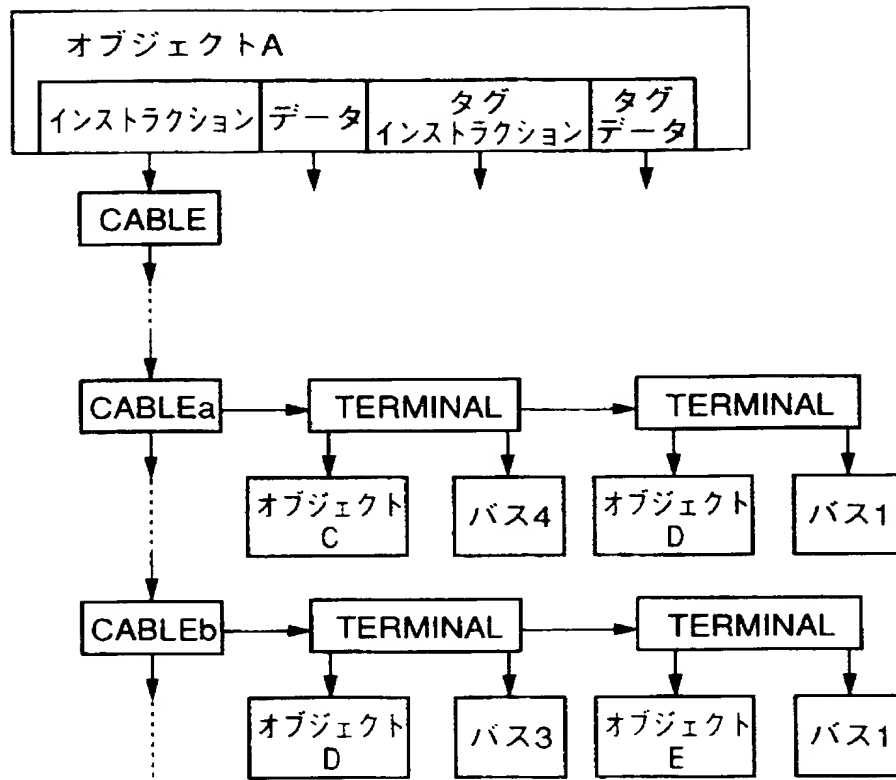
【図85】



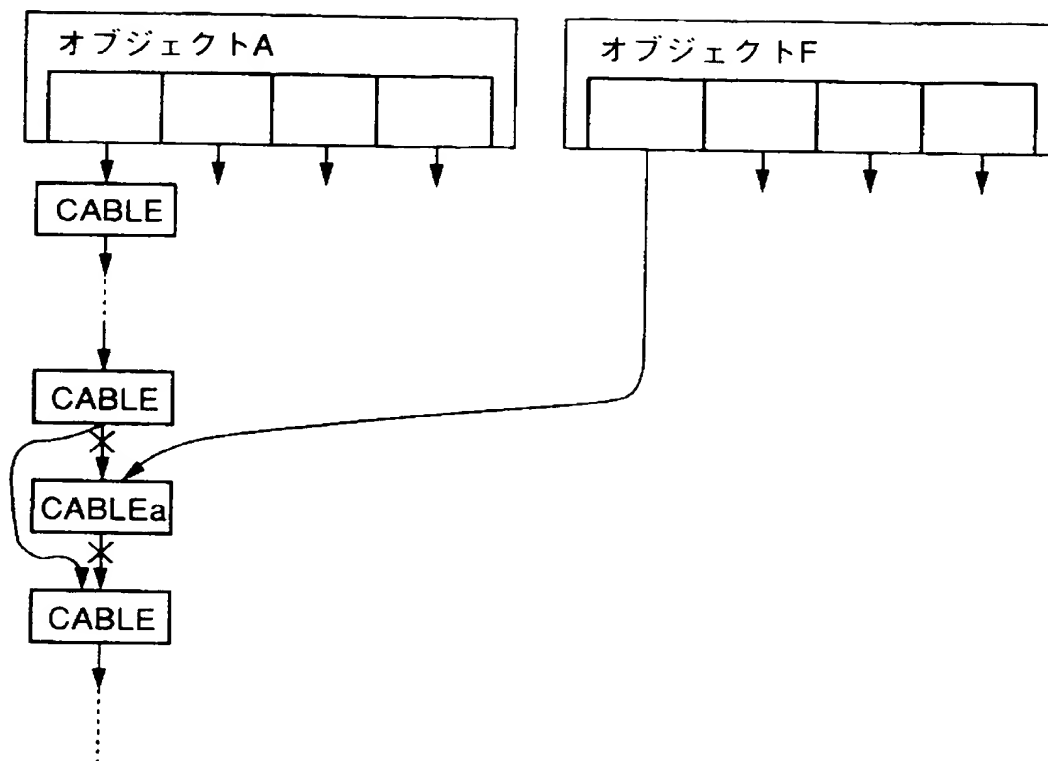
【図 86】



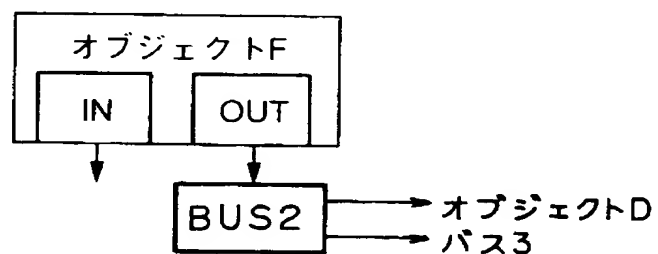
【図87】



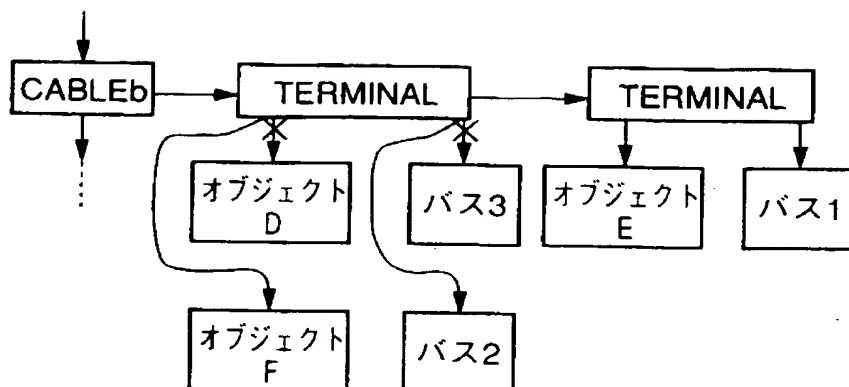
【図88】



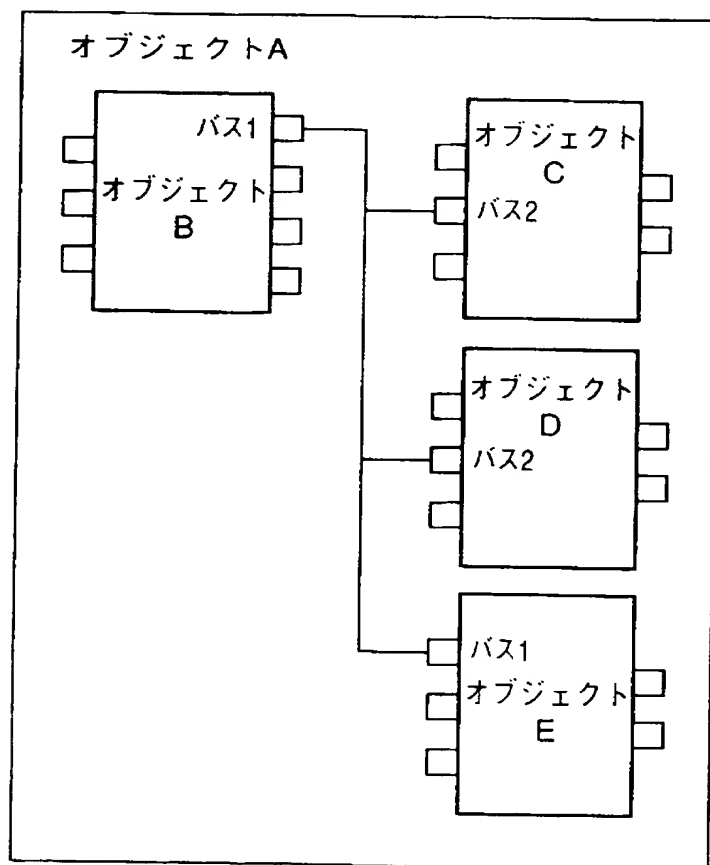
【図89】



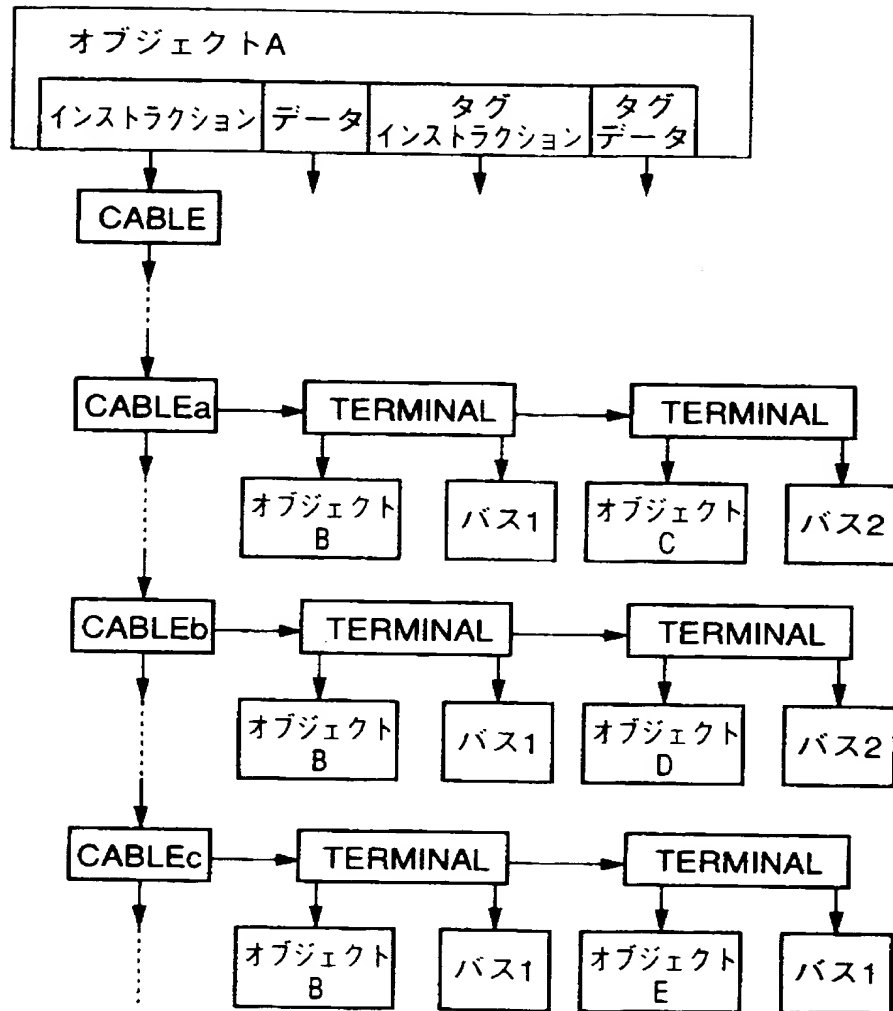
【図90】



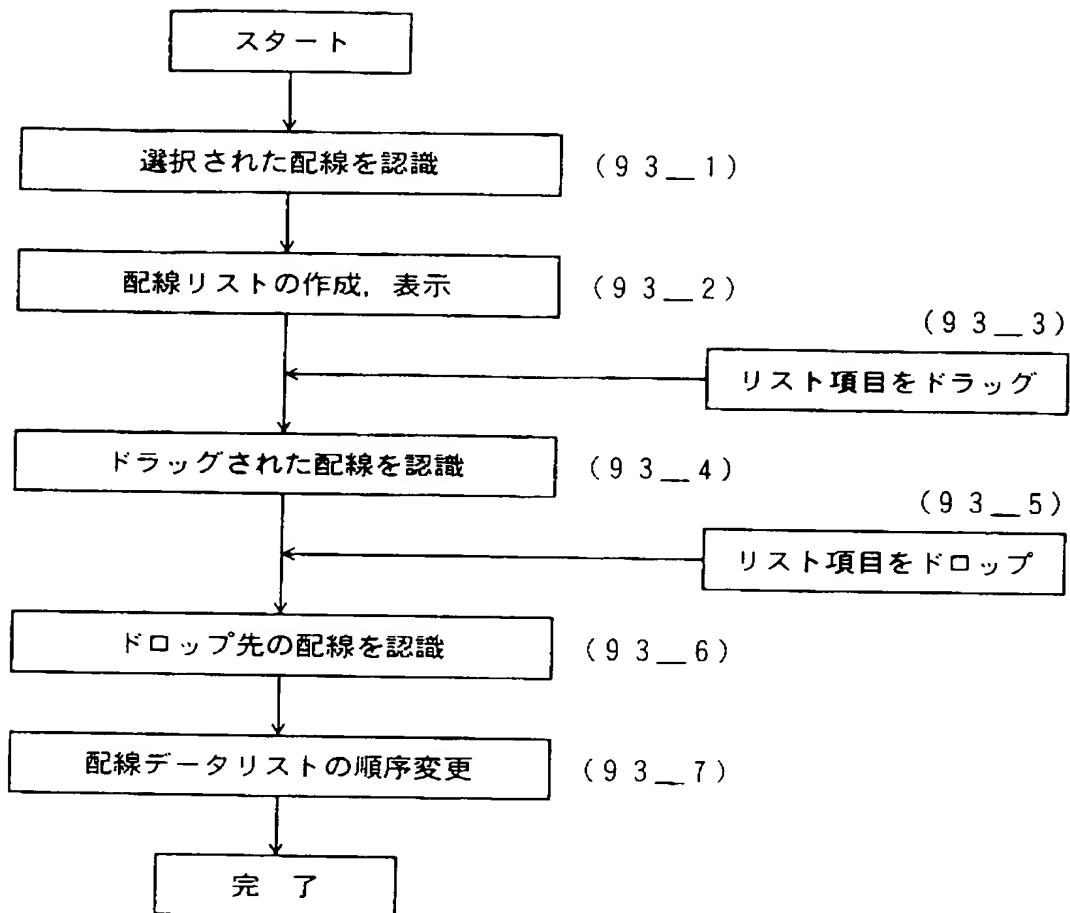
【図91】



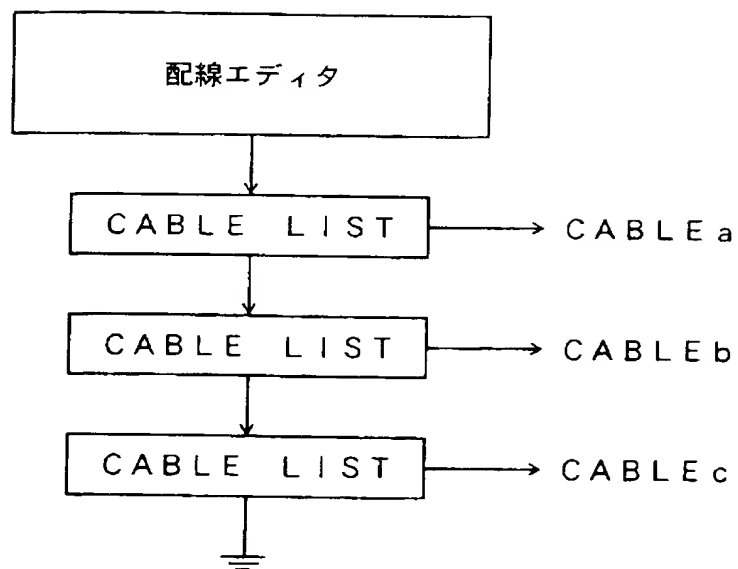
【図92】



【図93】



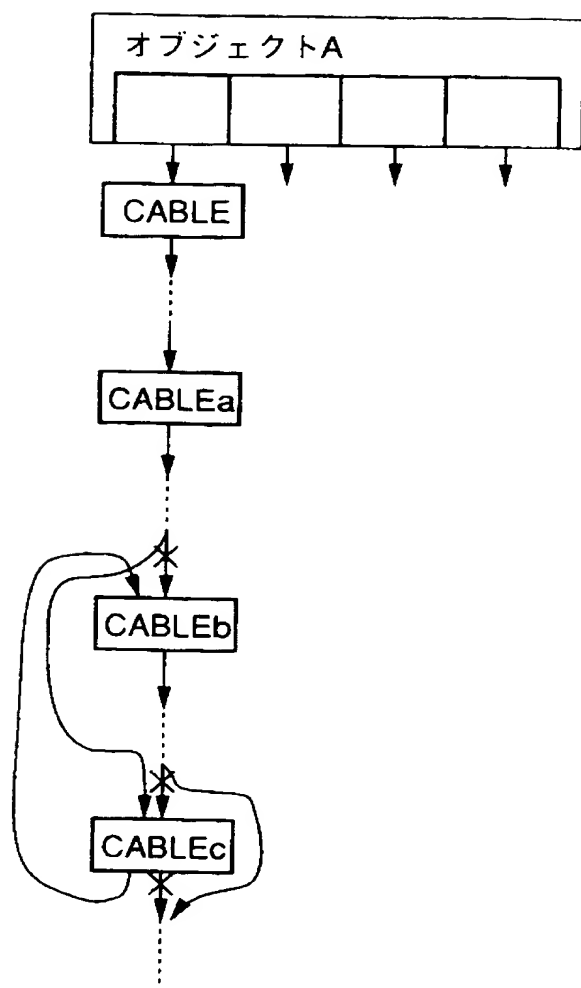
【図94】



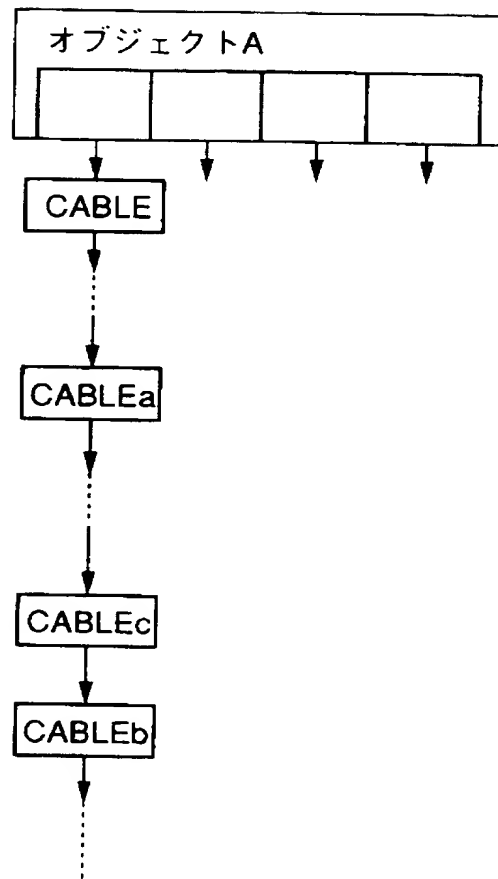
【図95】

オブジェクトB : バス1	オブジェクトC : バス2
オブジェクトB : バス1	オブジェクトD : バス2
オブジェクトB : バス1	オブジェクトE : バス1

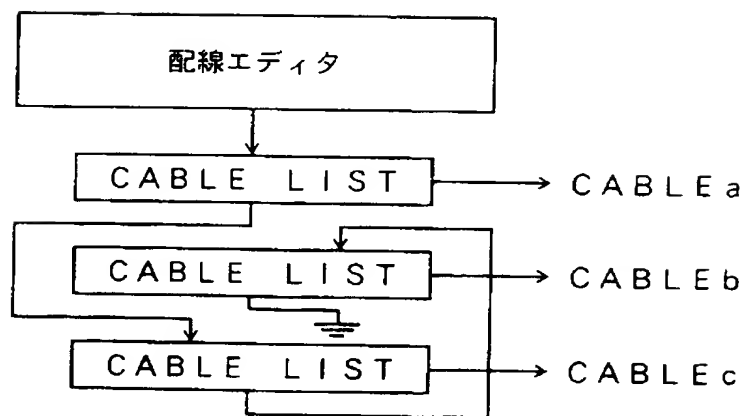
【図96】



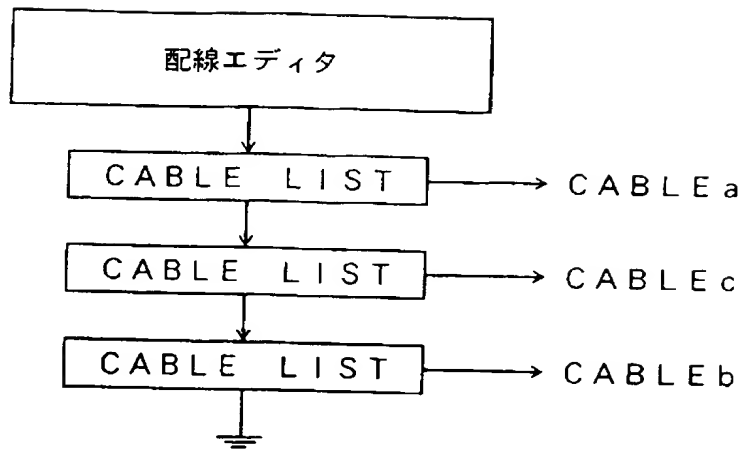
【図97】



【図98】



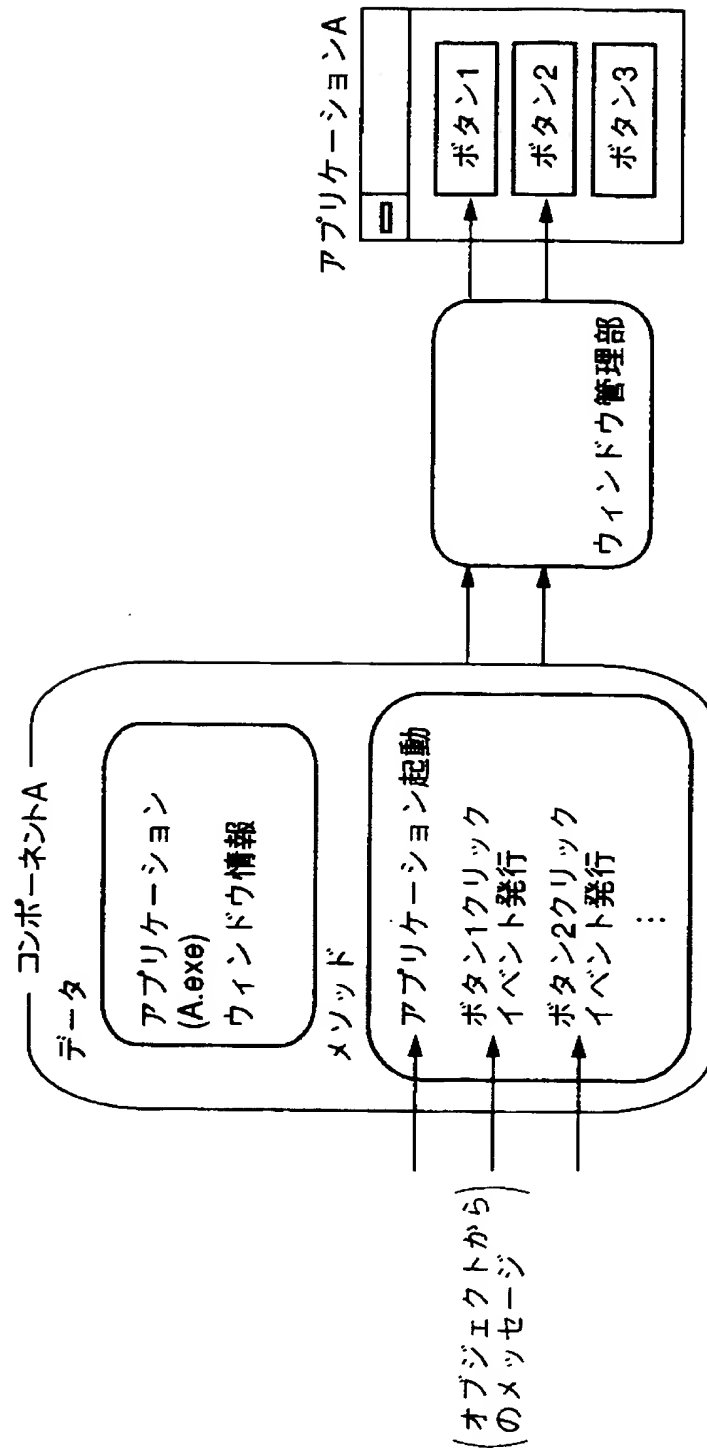
【図99】



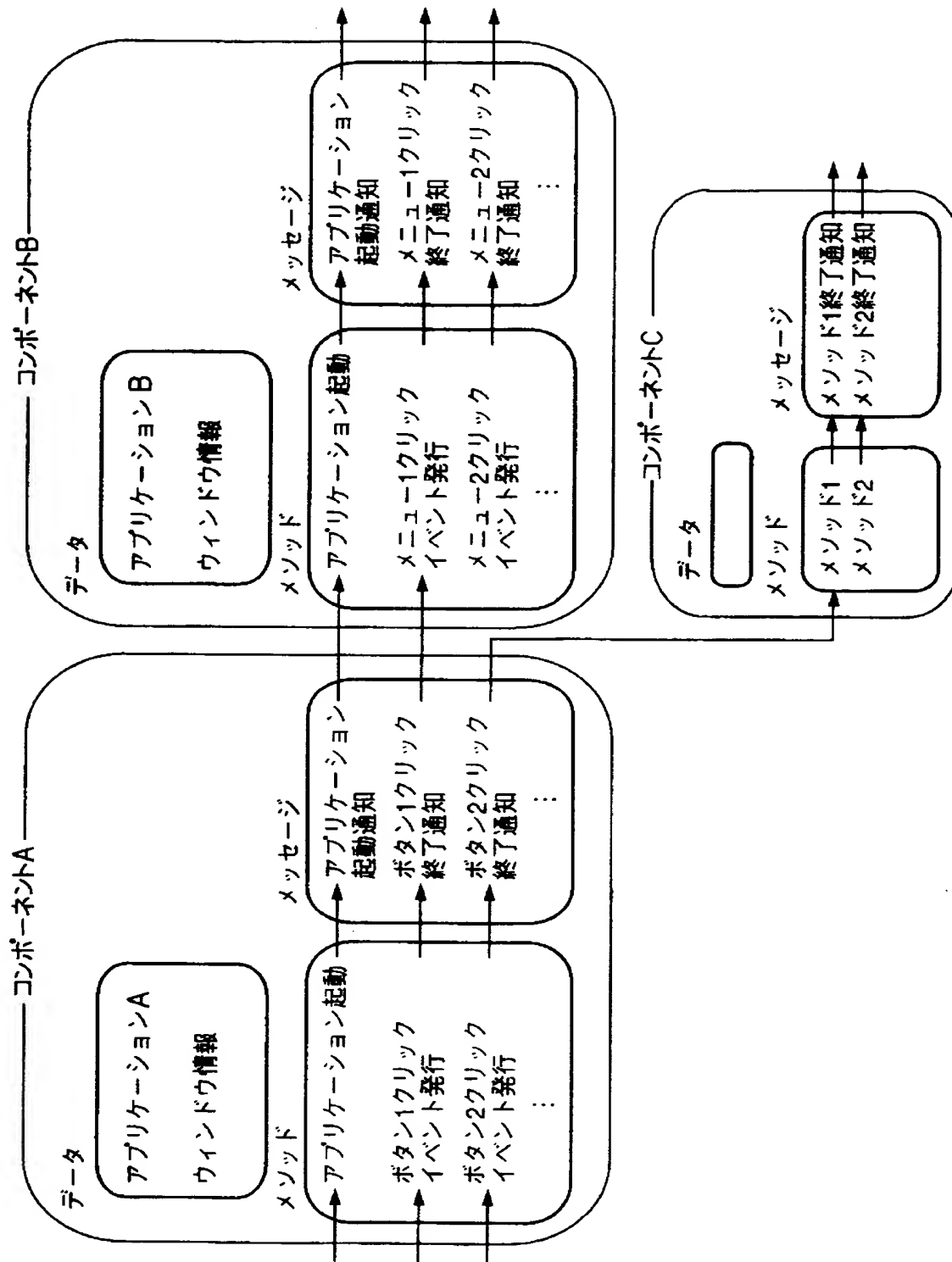
【図100】

オブジェクトB：バス1	オブジェクトC：バス2
オブジェクトB：バス1	オブジェクトE：バス1
オブジェクトB：バス1	オブジェクトD：バス2

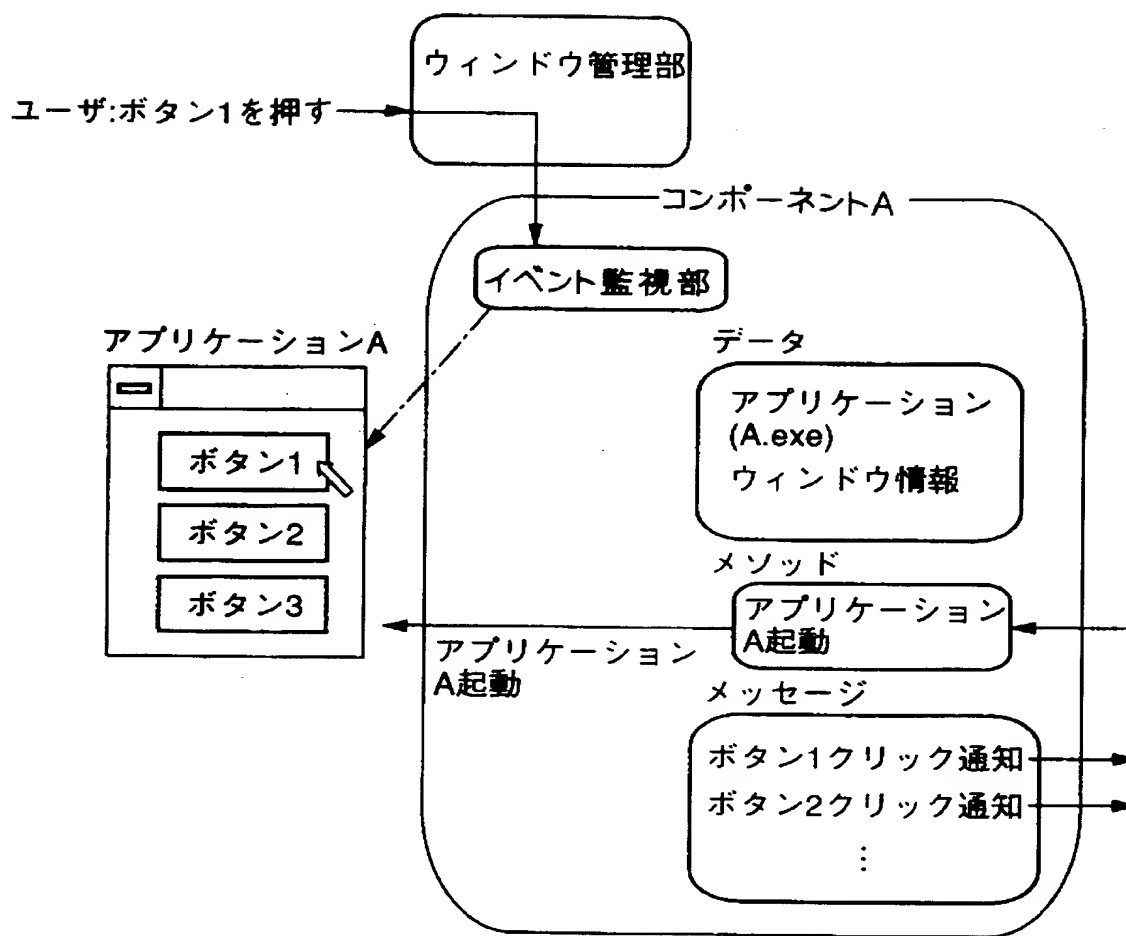
【図101】



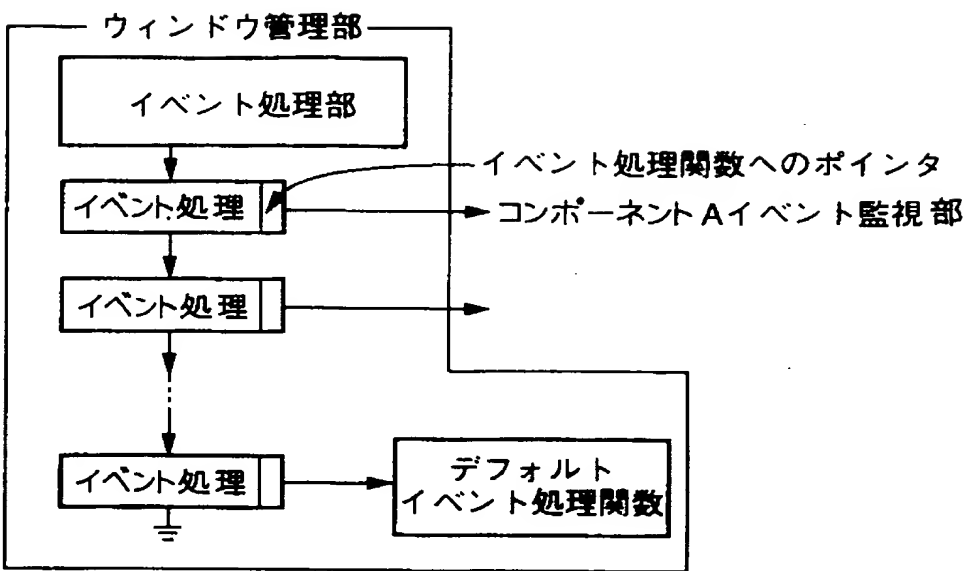
【図102】



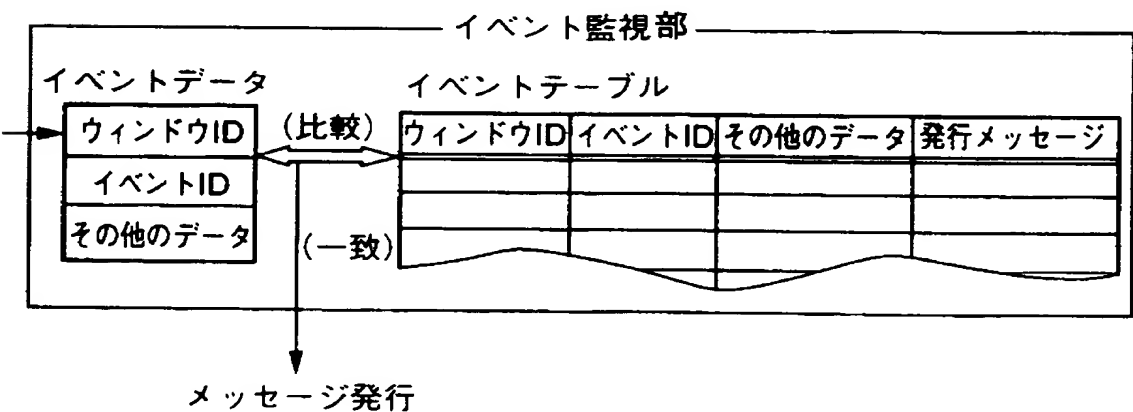
【図103】



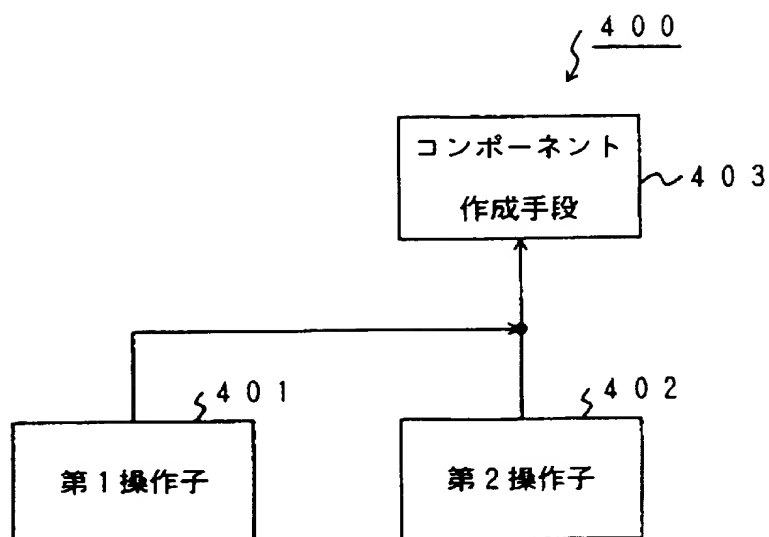
【図104】



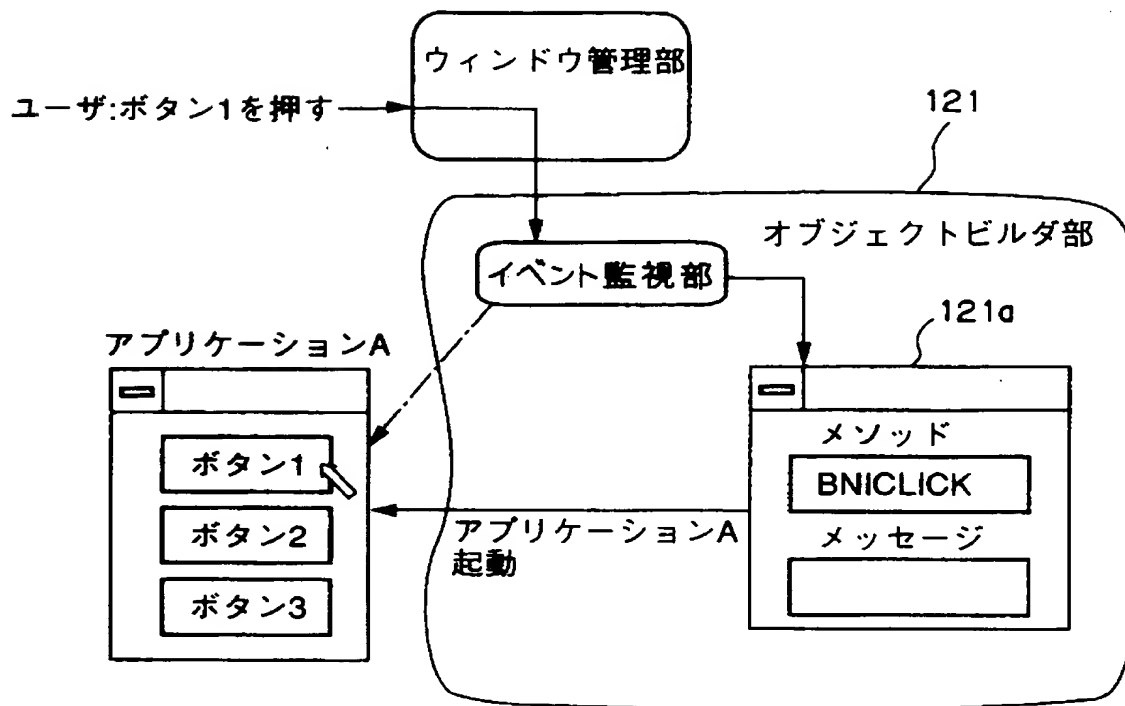
【図105】



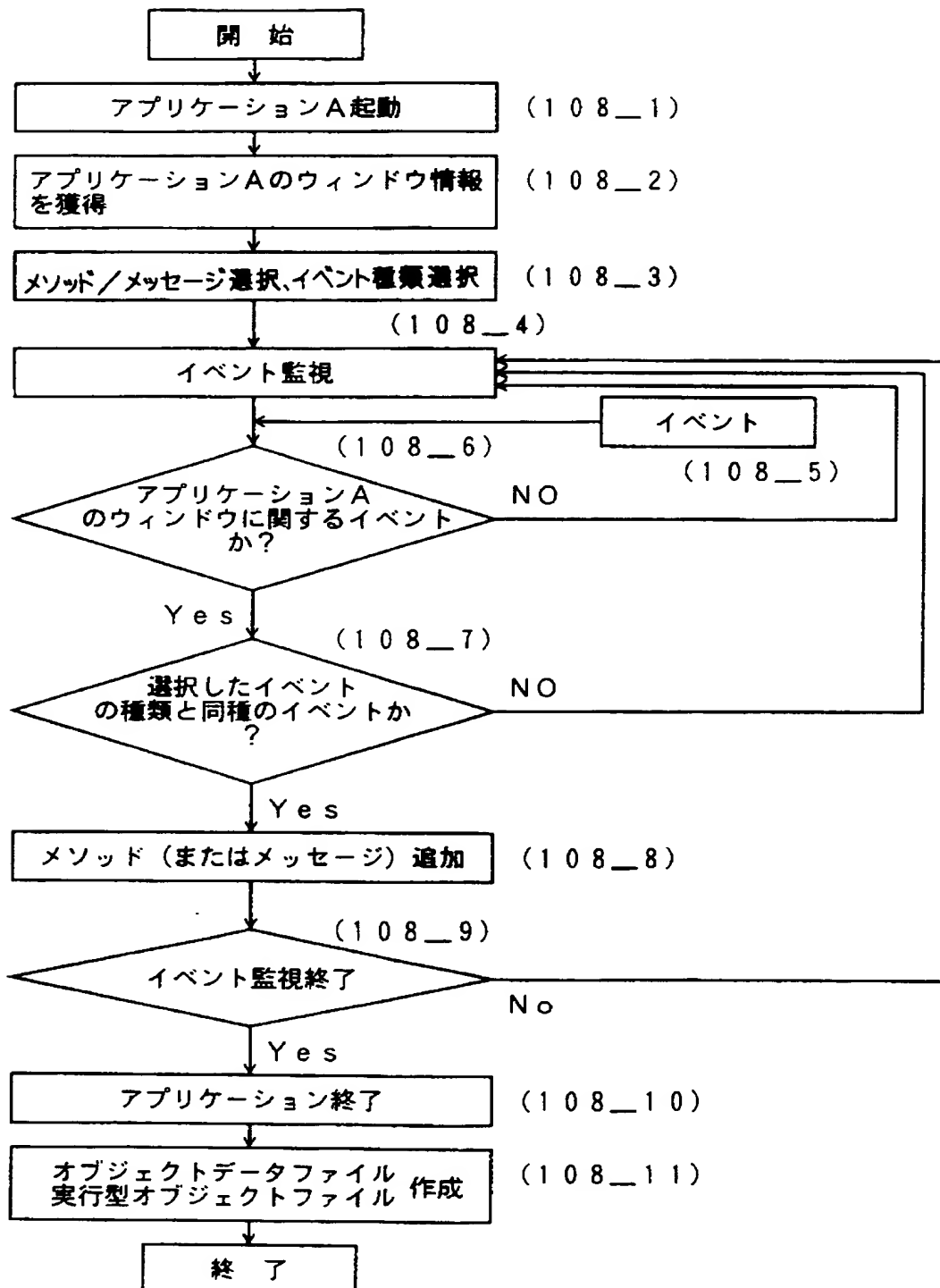
【図 1 0 6】



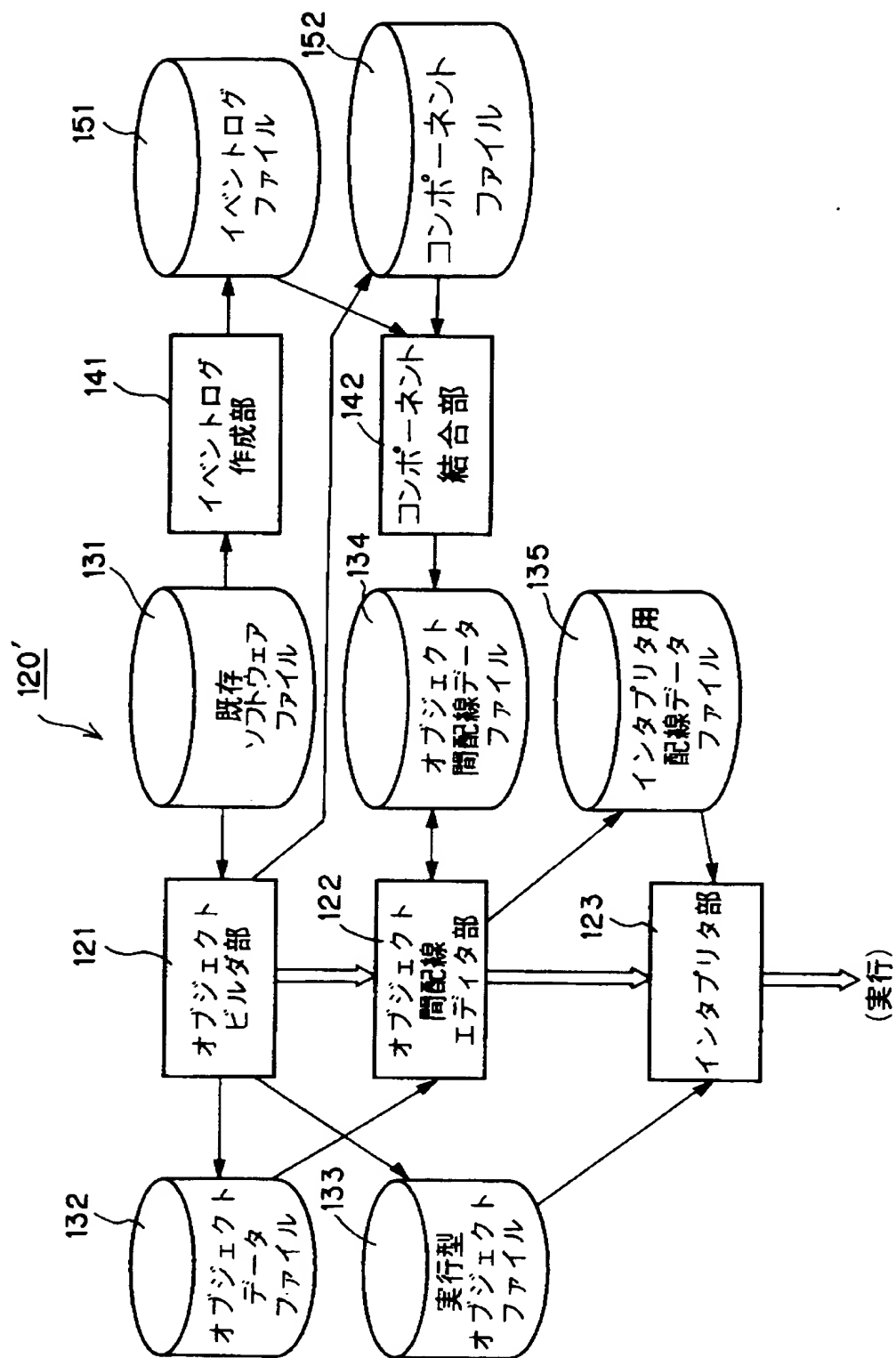
【図107】



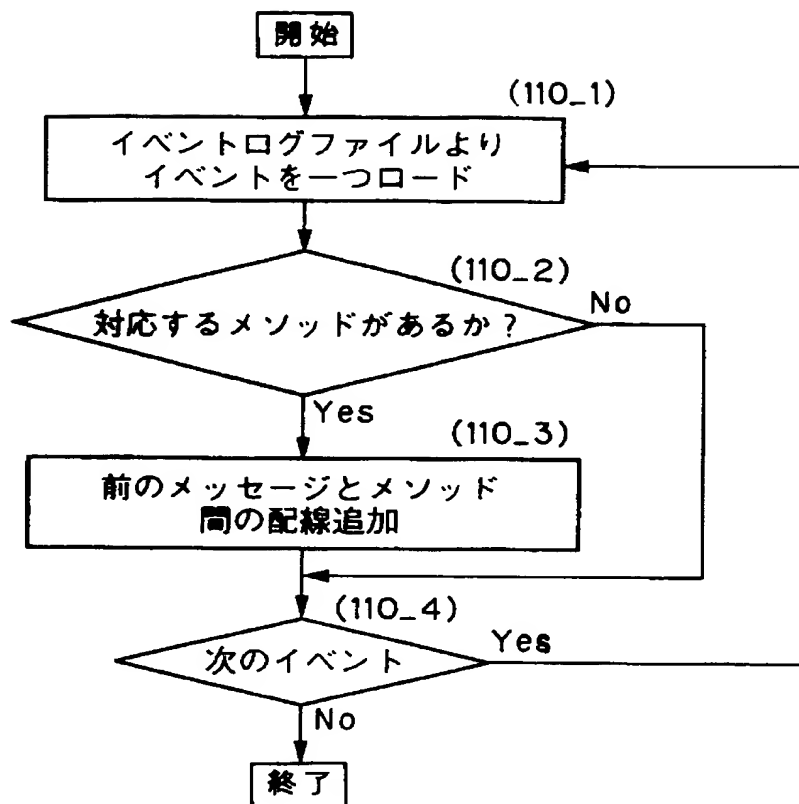
【図108】



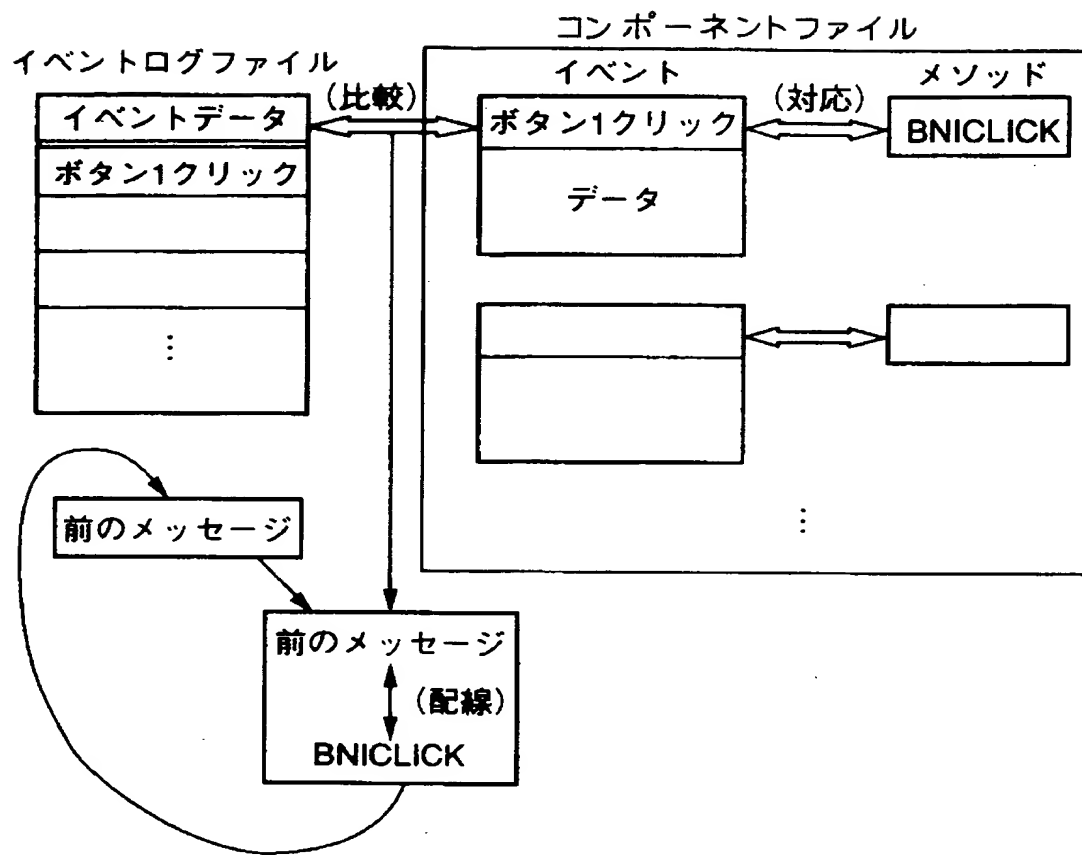
【図 1 0 9】



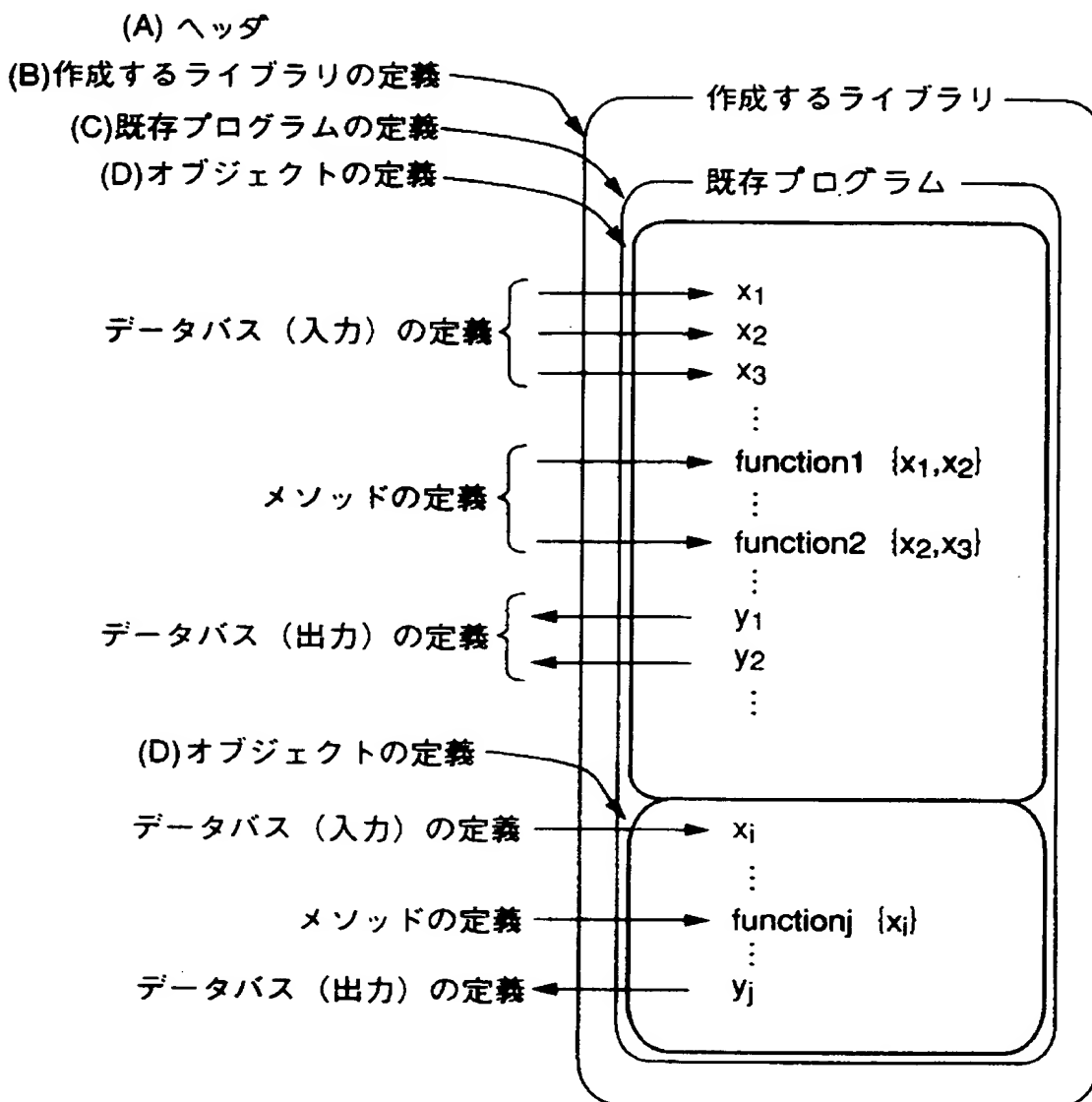
【図110】



【図111】



【図112】



【図113】

	項目	キーワード	備考
(A)	プロジェクト	LSIBuilderProject	
	プロジェクト名	LSIBuilderProjectName	
	コンパイラシステムのバス	MSVCRoot	
	First sightシステムのバス	CoreRoot	
	ユーザ領域のバス	UserRoot	
(B)	アーカイブ定義	Archives	
	アーカイブ名	ArchivesName	
	オリジナルソフトウェアLIBのバス	LibPath	
	オリジナルソフトウェアDLLのバス	DllPath	
(C)	作成するライブラリの名前	LibName	
	コンパイルモード	Debug	
	#define やtypedefの定義	Header	
(D)	LSI定義	LSI	
	LSI名	LSIName	
	LSIの色	Color	RGBの3色 (0-255)
	データバス	DataBus	
	データ収集プロセス名	ProcessName	
	データバスのバス名	Name	
	変数の型	VariableType	
	データ収集プロセス	Process	関数のコード
	DefineConnectorに直接展開	Inline	
	入力と出力の区別	IO	input又はoutput
	バスの色	Color	
	インストラクション	InstBus	
	インストラクションバスのバス名	Name	
	エントリポイントの関数名	ProcessName	
	戻り値の意味	ReturnValue	zero又はnonzero 又は数字
	インストラクションプロセス	Process	関数のコード
	Cmdかどうか	Cmd	yes/no
	CmdまたはCommandに直接展開	Inline	
	バスの色	Color	
	グローバル変数	Variables	
	(LSI内部のグローバル変数 (C++クラスのメンバ変数に相当))		
	#defineやtypedefの定義	Header	
	初期化処理	Initialize	関数のコード
	コンストラクタ	Constructor	関数のコード
	デストラクタ	Destructor	関数のコード

【図114】

ファイル 編集 表示 メイク ヘルプ																																									
<ul style="list-style-type: none">○project 名<ul style="list-style-type: none">○information○archives 名<ul style="list-style-type: none">○information○LSI 名<ul style="list-style-type: none">○information○instruction○data○LSI 名<ul style="list-style-type: none">○information○instruction○data○archives 名<ul style="list-style-type: none">○information○LSI 名<ul style="list-style-type: none">○information○instruction○data○LSI 名<ul style="list-style-type: none">○information○instruction○data	<div>LSI情報</div> <table border="1"><thead><tr><th>LSI名</th><th>LSIの色</th><th>define...</th><th>...</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td></tr></tbody></table> <div>インストラクションバス一覧表：</div> <table border="1"><thead><tr><th>バス名</th><th>エントリ...</th><th>戻り値...</th><th>...</th></tr></thead><tbody><tr><td>...</td><td></td><td></td><td></td></tr><tr><td>...</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></tbody></table> <div>データバス一覧表：</div> <table border="1"><thead><tr><th>バス名</th><th>変数の型</th><th>入力と出力...</th><th>...</th></tr></thead><tbody><tr><td>...</td><td></td><td></td><td></td></tr><tr><td>...</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></tbody></table>	LSI名	LSIの色	define...	...					バス名	エントリ...	戻り値...								バス名	変数の型	入力と出力...							
	LSI名	LSIの色	define...	...																																					
	バス名	エントリ...	戻り値...	...																																					
	...																																								
	...																																								
	バス名	変数の型	入力と出力...	...																																					
	...																																								
	...																																								

【書類名】 要約書

【要約】

【課題】 本発明は、オブジェクト指向プログラミングに関し、ソフトウェアの再利用性を高め、かつ実行速度を向上させる。

【解決手段】 オブジェクトAのデータ格納領域へのポインタが配列されてなるデータ要素リストと、オブジェクトBのポインタ格納領域へのポインタが配列されてなるポインタ要素リストを生成し、それらデータ要素リストとポインタ要素リストを結合することにより、オブジェクトAのデータをオブジェクトBから直接に参照できるようにする。

【選択図】 図 1 3

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】
【識別番号】 000005223
【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号
【氏名又は名称】 富士通株式会社
【代理人】 申請人
【識別番号】 100094330
【住所又は居所】 東京都港区西新橋三丁目 3 番 3 号 ペリカンビル 4
階 小杉・山田国際特許事務所
【氏名又は名称】 山田 正紀

特願平 0 9 - 1 7 1 7 8 2

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

1. 変更年月日

1 9 9 6 年 3 月 2 6 日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名

富士通株式会社